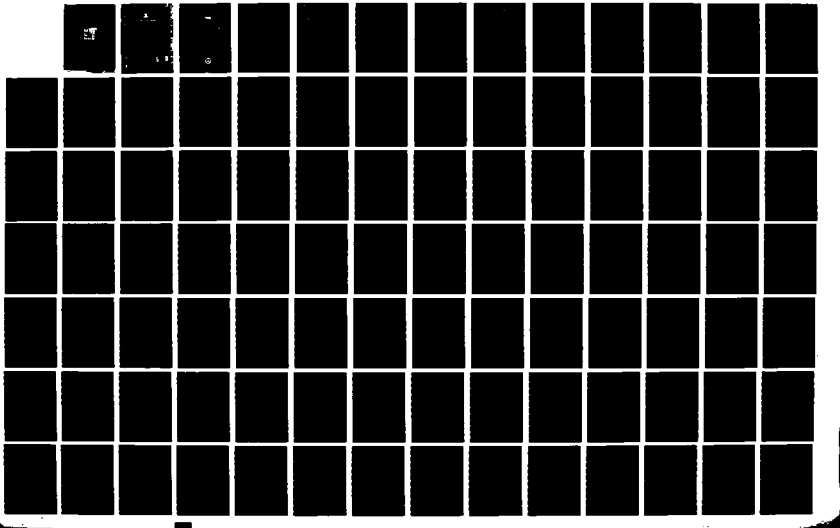
AD-A157 129 VLSI WORKSHOP AND PROJECT: FEBRUARY-NOVEMBER 1982(U) 1/2
WEAPONS SYSTEMS RESEARCH LAB ADELAIDE (AUSTRALIA)
J HAYWARD SEP 84 WSRL-0374-TR

UNCLASSIFIED F/G 9/5 NL




1.1
1.25

WSRL-0374-TR

AR-004-105

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

WEAPONS SYSTEMS RESEARCH LABORATORY

DEFENCE RESEARCH CENTRE SALISBURY
SOUTH AUSTRALIA

TECHNICAL REPORT
WSRL-0374-TR

FINAL REPORT ON VLSI WORKSHOP AND PROJECT: FEBRUARY – NOVEMBER 1982

J. HAYWARD

Approved for Public Release

SEPTEMBER 1984

COPY No. 42

AR-004-105

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

WEAPONS SYSTEMS RESEARCH LABORATORY

TECHNICAL REPORT

WSRL-0374-TR

FINAL REPORT ON VLSI WORKSHOP AND PROJECT: FEBRUARY - NOVEMBER 1982

J. Hayward

SUMMARY

At regular intervals, the CSIRO VLSI Program coordinates the production of multiproject very large scale integrated circuit chips, in n-mos technology, using the Mead-Conway design method. The author attended the first Workshop conducted in support of this activity, and designed a successful digital circuit project which became part of the first such chip. This report describes the Workshop, the project, the circuit design, the performance estimation and measured performance. It also covers the difficulties encountered in transfering the computer software design tools, supplied by the Program in a form suitable for the VAX computer operating system, to the main-frame IBM 3033 system in use at DRCS.

POSTAL ADDRESS: Director, Weapons Systems Research Laboratory,
Box 2151, GPO, Adelaide, South Australia, 5001.

TABLE OF CONTENTS

		Page
1.	INTRODUCTION	1
2.	PROCESSES FOR REALISING DIGITAL LOGIC	2
	2.1 General	2
	2.2 Resume of ideal digital logic	2
	2.3 Bipolar process	3
	2.4 N-mos process	3
	2.5 C-mos process	3
	2.6 Other processes	3
3.	THE INSTRUCTOR'S WORKSHOP	4
	3.1 Introduction	4
	3.2 Course organisation	5
	3.3 Course content	5
	3.4 Reference material issued to students	6
	3.5 Facilities available	6
	3.6 Computer programs available	7
	3.7 Program documentation	7
	3.8 Conduct of course	8
	3.9 Projects	8
4.	SCHEDULE FOR MULTIPROJECT CHIP (AUSMPC 5/82)	8
	4.1 General	8
	4.2 Project design phase	8
	4.3 Project merging and chip allocation	9
	4.4 Data format conversion	9
	4.5 Mask writing	9
	4.6 Mask making	9
	4.7 Wafer fabrication	9
	4.8 Chip packaging	10
	4.9 Chip delivery	10

	Page
4.10 Photographs	10
4.11 MPC documentation	10
4.12 Chip testing	10
4.13 Design workshops	10
5. PROCESS PARAMETERS FOR CHIP DESIGN	11
5.1 N-mos process parameters	11
5.2 Derived parameters for actual wafers	11
6. DESIGN RULES FOR AUSMPC 5/82	12
6.1 Dimensions	12
6.2 N-mos design rules	12
6.3 CIF rules	13
6.4 Library cell status	14
7. VAX TO IBM LIBRARY CONVERSION	15
7.1 General	15
7.2 Choice of host computer	15
7.3 Tape conversion problems	16
7.4 Contents of the tape	17
7.5 Reading a particular file	17
7.6 Reformatting the file for display	17
7.7 Reformatting the file into the original text	17
7.8 Conversion of PASCAL source programs	18
7.9 Conversion of other text files	18
8. IBM TO VAX DATA TRANSFER VIA MAGNETIC TAPE	18
8.1 General	18
8.2 Tape format	18
8.3 Writing the tape	19
8.4 Data transfer	19
9. USING THE UTILITY PROGRAMS TO PRODUCE DESIGN CODE	19

Page

10. THE SERIAL DATA MATRIX TRANSPOSER AS AN MPC PROJECT 19

10.1 Binary digital communication 19

10.2 Burst error elimination 20

10.3 Multiplexing of data sources 20

10.4 Organisation of the transposer stores 21

10.5 Choice of organisation for the MPC 21

11. LOGICAL SYNTHESIS OF THE SERIAL DATA MATRIX TRANSPOSER 22

11.1 Two-dimensional shift-register array 22

11.2 Two-dimensional shift-register cell 22

11.3 Complete Matrix Transposer 23

11.4 Compliance with functional requirements 24

12. DESIGN OF SHIFT-REGISTER ARRAY 24

12.1 General considerations 24

12.2 Design of the shift-register cell 25

12.3 Size of arrays for MPC 26

12.4 Ancillaries to the array 26

13. DESIGN OF THE DATA OUTPUT SELECTOR 26

13.1 General 26

13.2 Design of the PLA 27

13.3 Ancillaries to the PLA 27

14. DESIGN OF THE CLOCK SELECTOR SWITCH 28

14.1 General 28

14.2 Design of the pass-transistor network 28

15. DESIGN OF THE CLOCK SPLITTER 29

15.1 General 29

15.2 Clock splitter design 30

15.3 Clock skew and delay elements 30

15.4 Clock-Splitter BELLE-Code 31

		Page
16.	PROJECT-DEFINING BELLE-CODE	31
	16.1 General	31
	16.2 Library cells	31
	16.3 Element flow sequence	32
	16.4 Chip area	33
17.	ESTIMATION OF CHIP CURRENT CONSUMPTION	33
	17.1 General method	33
	17.2 Saturation current	33
	17.3 Charging current	34
18.	ESTIMATION OF OPERATING SPEED	34
	18.1 General	34
	18.2 Model of n-mos FET transistor	35
	18.3 Clock Splitter delays	36
	18.4 Shift register array	37
	18.5 Clock selector switch	39
	18.6 Output PLA	40
	18.7 Summary	40
19.	TESTING THE CHIP	40
	19.1 Built-in test features	40
	19.2 Functional Test Assembly	41
	19.3 Functional test procedure	42
	19.4 Test equipment used	42
	19.5 Chips under test	43
	19.6 Test results	43
	19.7 Discussions of results	43
20.	CONCLUSION	44
21.	ACKNOWLEDGEMENT	44
	REFERENCES	45

Page

LIST OF APPENDICES

I	READING FILE FROM VAX TAPE INTO IBM DATASET	46
II	REFORMATTING THE BLOCK-DATA FOR DISPLAY	48
III	REFORMATTING THE BLOCK-DATA TO LINE DATA	49
IV	MODIFICATIONS TO PASCAL SOURCE PROGRAMS	52
V	TRANSFER OF DATA FROM IBM TO VAX VIA MAGNETIC TAPE	54
VI	USING THE VLSI UTILITY PROGRAMS PLAGEN, GETSYMBOLS AND BELLE	56
VII	PROJECT-DEFINING BELLE-CODE	57

LIST OF TABLES

1.	N-MOS PROCESS PARAMETERS	71
2.	CONTENTS OF WORKSHOP MAGNETIC COMPUTER TAPE	72
3.	DATA-FLOW AS A FUNCTION OF CONTROL SIGNAL LEVELS	72
4.	DEPLETION MODE CHANNEL SIZE FOR BASIC ELEMENTS	73
5.	DEPLETION MODE CHANNEL SIZE FOR A SHIFT REGISTER ARRAY	73
6.	DEPLETION MODE CHANNEL SIZE FOR A CLOCK SPLITTER	73
7.	DEPLETION MODE CHANNEL SIZE FOR THE CLOCK SELECTOR SWITCH	74
8.	DEPLETION MODE CHANNEL SIZE FOR THE DATA OUTPUT SELECTOR	74
9.	ESTIMATE OF CAPACITY DRIVEN BY CLOCK WAVES	74
10.	CLOCK SPLITTER ESTIMATED TIME SEQUENCE	74
11.	SHIFT REGISTER CELL TRANSIENT RESPONSE	75
12.	SHIFT REGISTER CELL TRANSIENT RESPONSE	76
13.	DYNAMIC TEST SPECIFICATION	77
14.	MEASURED CHIP CURRENTS (mA)	77

LIST OF FIGURES

1. Inverter, NOR and NAND gates
2. Memory element
3. Complementary inverter
4. X & Y shift register array

5. Shift register cell
6. Clock-derived control signals
7. Logic diagram of project
8. Shift register cell
9. Floor plan of project
10. Clock Selector Switch - stick diagram
11. Standard cell PADCLOCKBAR
12. Clock Splitter
13. Clock Splitter physical arrangement
14. FET transistor model
15. FET gate model (R = 4)
16. FET gate model (R = 8)
17. Clock Splitter circuits
18. Clock Splitter estimated delays
19. Non-Inverting Super-Buffer
20. Equivalent circuit of array cell
21. Simulation of shift register transient (0 to 1)
22. Simulation of shift register transient (1 to 0)
23. Equivalent circuits
24. Chip connections and internal logic
25. Logic diagram of functional test set up
26. Waveforms for functional test set up

1. INTRODUCTION

For many years, small scale integrated circuits (consisting of tens of electronic digital logic elements and inter-connecting wires, chemo-photographically deposited on small wafers of pure silicon, and contained within a small metal or ceramic package complete with pins for external connection) have been available and widely used.

More recently, due to improved resolution of the chemo-photographic process and better understanding of the physics of the logic elements, the element size has been greatly reduced, resulting in tens of thousands of logic elements fitting on to the same size wafer. This level of integration has been named VLSI (Very Large Scale Integration).

Two distinct trends have emerged in the use of VLSI. Firstly, IC Manufacturers have produced for sale, at remarkably low prices, "standard" functions such as micro-processors and random-access-memories. These make full use of the techniques to cram the maximum amount of logic in a given space, and are designed by "professional" VLSI designers, using extensive experience of what their own production methods can achieve with a reasonable yield of working chips.

Secondly, US Universities and Research Laboratories have combined to produce a set of simply understood design rules which, if followed, permit "amateur" VLSI designers to design their own particular logic functions, at a significantly lower element density and operating speed, but with a high probability of successful yield when produced by virtually any Manufacturer's production methods. Additionally, they have produced a "standard language" called CIF (Caltech Intermediate Form). A sequence of statements in this language, located in computer memory or on magnetic tape, can define a VLSI design in a manner comprehensible to both the original designer, and the chemo-photographic processor which produces the hardware chip. This has made possible a complete separation of the functions of design and manufacture, as a result of which several smaller firms in the US now specialise in the manufacture of out-of-house designs, and have been nick-named Silicon Foundries.

This second trend has been recognised by a number of Government and private organisations in Australia as offering significant possibilities for the development of locally required, special purpose integrated circuits of high logical complexity, small size and power consumption, and at a not too exorbitant cost for small to moderate quantities. The Defence Science and Technology Organisation (DSTO) is one such organisation, being charged with the development and modification of weapon and communication systems for the Armed Forces.

Would-be VLSI designers require some training and practice in the use of the design rules, followed by the opportunity to test the performance of one or two versions of actual hardware produced from their practice designs. The cost of this hardware becomes reasonable only if one wafer of IC chips can contain the designs of a large number of students. Fortunately this is possible under the multi-project chip program, invented by the same combination of US Universities and Research Laboratories.

In 1980, the CSIRO Division of Computing Research set up the VLSI Program (located in Adelaide, South Australia, and headed by Dr Craig Mudge) to become expert in the techniques of VLSI function design, and to stimulate interest and application within Australian engineering laboratories and institutes of tertiary education. To this end, the Program has sponsored a series of multi-project chips, at intervals of four to six months, the first closing in May 1982. It also organised an Instructor's Workshop (held in Adelaide in February 1982) for key Australian designers and lecturers who could be expected to spread the message among students and associates.

was selected to attend the Workshop as the only DRCS representative, and to ntribute a design for inclusion in the first multi-project chip. This report scribes my reactions to the Workshop, difficulties encountered in mastering the sign tools, details of the submitted design, methods used to test the hardware ceived from the Foundry, and the actual measured performance of what proved to a completely successful implementation.

2. PROCESSES FOR REALISING DIGITAL LOGIC

2.1 General

There are several distinct processes available for the realisation of digital integrated circuit logic on a silicon substrate, eg bipolar, p-mos, n-mos and c-mos. Each has advantages and disadvantages. Some important attributes include speed of operation, power consumption, yield and cost, as well as flexibility in the provision of a range of logic elements, simplicity of the essential design rules, and the ability to provide for cross-overs in the interconnection "wires".

2.2 Resume of ideal digital logic

The basic element is the logic gate: a cell with one output line and several input lines. The line signals have two possible levels (of some physical quantity, usually voltage) called for convenience "High" and "Low" (H and L), although there may be no direct relation between the level identification and the magnitude (or polarity) of the physical levels. The level on the output line is logically related to the levels on the input lines through the well known AND, OR and INVERT functions, and their combinations. It is usual (although not essential) for the physical-logical level association to be the same at all inputs and outputs, to permit ready interconnection of gates. To this end also, it is necessary that a single output must be capable of feeding many inputs in parallel, without significant loading effects on the physical level (fan-out capability).

Figure 1(a) shows the ideal form of INVERT gate, consisting of the current source, C, the clamping diode, D, (connected to the H-level voltage rail, V) and the input-controlled switch, S. The switch, S, is understood to be closed (passing current) when its controlling signal is at the H-level. The L-level is, of course, the voltage represented by the earth symbol. Figure 1(b) shows the equivalent logical symbol for the gate.

Figure 1(c) shows how several switches can be combined to produce a NOR-gate, symbolised in figure 1(d). Parts (e) and (g) of figure 1 show two ways of realising the NAND-gate, symbolised in figure 1(f).

The basic static memory element is shown in figure 2. This is effectively two INVERT elements cascaded, with the signal input via S_1, and with feedback from the output to the input via S_2. The capacitor, X, provides short term memory in case both switches are open, momentarily, at the end of the WRITE-pulse.

For dynamic memory, S_2 is omitted. X holds the data until its charge leaks away through the open S_1, or through the control line of S_3. The data may be kept indefinitely if it can be refreshed periodically via S_1.

For ideal complementary logic, the current source and clamping diode are replaced by a "complementary" switch, S_1 in figure 3, between the H-rail and output line. S_1 is closed when S_2 is open, and vice versa. The input signal may require logical inversion, and/or level shifting, before application to the control line of S_1.

2.3 Bipolar process

The bipolar process is capable of producing NPN transistors (for switches diodes and resistors (for current sources). It is fast but uses more power and chip area than some other processes.

The complementary bipolar process is also capable of producing PNP transistors (for the complementary switches). Its power consumption is significantly lower, but its area is still rather large.

Both types have complex design rules. Both exhibit relatively large leakage through open switches, and their control lines, ruling out the use of dynamic memories.

2.4 N-mos process

The n-mos process is capable of producing only n-channel mos-fet transistors (both depletion and enhancement types). No resistors or diodes are possible. Enhancement type transistors are used for the normal switches. Depletion type transistors, with gate tied to source are used as current source resistors (between the V-rail and output lines).

The open switch leakage and control line leakage of the transistors are extremely low, however, permitting dynamic memory operation for long periods using only the (very small) intrinsic capacity between the control gate and the conduction channel (hundredths of a pico-farad).

The process is very well understood; design rules are simple; and the area of the chip can be very small. Power consumption is rather large. The speed of the logic circuitry is only moderate, but it is fast enough for most general purposes.

This process is available in Australia from AWA Ltd, but only on a laboratory basis: no mass production techniques are used.

It is the process used for most of the recent commercial successes (micro-processors, RAM memories, etc) but is being superseded by the c-mos process.

2.5 C-mos process

This, the complementary version of n-mos, has the advantage of higher speed and reduced power consumption due to the use of a complementary transistor switch to replace the current source. The chip area is rather similar to n-mos. The design rules are more complex, and, of course, the production problems are greater.

AWA Ltd is working towards providing a Foundry service for this process, which is likely to become the industry standard.

2.6 Other processes

There are other processes, and there are variations within each process. Only n-mos is presently standardised sufficiently for amateur use, although c-mos is expected to supersede it in the not too distant future.

N-mos, using the design rules prepared by Mead and Conway(ref.1), was the only process considered by the CSIRO Workshop, and the process specified for the CSIRO multi-project chips.

3. THE INSTRUCTOR'S WORKSHOP

.1 Introduction

ıe course was held at the CSIRO VLSI Program Offices, in the AMDEL group of ıildings, Frewville, South Australia. It took place over the three week ɜriod of 8 to 26 February 1982, inclusive. The course Director was r Craig Mudge, leader of the VLSI Program.

ıe main series of lectures was presented by Drs D.A. Pucknell and . Eshraghian of the University of Adelaide. Specialist lectures were resented by C. Mudge, M. Haskard, P. Maxwell, M. Paltridge, R. Clarke and . Phillips of the VLSI Program; G. Rigby of AWA Ltd; and I. Sutherland, a ɔnsultant from USA. Supplementary lectures were provided (via video tape) ɣ VLSI Technology Inc. These were recordings of certain lectures given to similar Workshop conducted in 1980 in the USA.

ıtorials on VLSI design were conducted by members of the VLSI Program based n their experience with paper designs. None of them at that stage had had design produced in hardware, using the techniques covered in the Workshop. he earlier tutorials were based on design problems set by the lecturers. he latter ones were centered upon the problem chosen by the individual tudent for his multi-project chip project.

he resident computer system (a VAX 11/780, owned by the VLSI Program) was ade available to students. Students had access to the electronic mail unction, to two basic programs for generating CIF-code, and to one basic rogram for printing out a design (held in CIF-code) in the form of a two imensional layout, via a multi-colour-pen graphic plotter. More sophisti- ated programs, held by the Program, and capable of checking conformance with esign rules, or of predicting the performance of the hardware, were not made vailable to students. This was in accordance with the conditions of release f the programs from the US to the Program. The running time of these programs as such that the VAX would have been grossly overloaded had they been avail- ble for student use.

ther VLSI Program facilities made available to students were the library, pecial purpose instruments (eg binocular microscopes, bed-of-nails chip esters) and copious photographs of commercial and amateur silicon chips.

wenty three students began the course, and one withdrew after the first day. ourteen were from teaching institutions and nine from industry and government epartments and authorities. Most seemed to be more conversant with computer cience than with electronic circuit design. My situation was the opposite ne. All students were intensely interested in the subject and were more than illing to work long hours. Students were issued with office keys giving 4 hour access to the building. The VAX computer ran 24 hours a day, and each tudent had a terminal. Conditions, then, were ideal (especially for inter- tate students, living near the campus) except for the constraint of nadequate office space: quite serious, considering the size of drawings to e prepared, and the number of references to be consulted while carrying out esigns.

he course was designed entirely around the n-mos process, which is simple to pecify and is a good introduction to VLSI. Perhaps 90% of the course-work as relevant to (or provides a basis for) an extension into c-mos design at ome later stage.

(d) the space, or character "¬", appear as dummy characters to fill out a block.

ppendix III lists the PASCAL program (and CLIST required to execute it) to ead the block-format from one dataset and to reformat it into a normal lined ext-type dataset. At the same time, it converts:

(1) lower case letters to upper case letters;

(2) "¢", "!" and "¬", to "(.", ".)" and "@";

(3) "{" and "}", to "(*" and "*)";

(4) long lines of text to two or more shorter ones.

ote the method of validating PASCAL strings, split into two lines by the eformatting program (described in the appendix).

.8 Conversion of PASCAL source programs

he source programs, BELLE, PLOTCIF, PLAGEN and GETSYMBOLS, were extracted rom the magnetic tape, HAYWAR, and reformatted as described in ub-sections 7.5 and 7.7. They were then compiled using the PASCALVS compiler n the IBM system, to flag incompatibilities. Some modifications were equired, and these are described in Appendix IV, for three of the programs.

he fourth, PLOTCIF, required more drastic modification to suit the plotting acilities available on the IBM system. The author has not addressed this roblem to date, although solutions have been produced by other VLSI esigners at DRCS.

.9 Conversion of other text files

11 other files on the magnetic tape (Table 2) were extracted as described, nd required no editing.

8. IBM TO VAX DATA TRANSFER VIA MAGNETIC TAPE

.1 General

he project design for the multi-project chip (in CIF-code) had to be returned o the VLSI Program VAX computer for testing and merging (see Sub-sections 4.2 nd 4.3). As the occurrence of a single error could be fatal, inter-computer ransfer via the MAIL facility was ruled out. Transfer by magnetic computer ape was the only medium considered adequate and four acceptable formats were pecified.

.2 Tape format

he format chosen was:

(a) fixed length card images on unlabelled tape;

(b) fixed logical record length of 80 bytes;

(c) ASCII characters;

(d) 9 tracks;

7.4 Contents of the tape

Information provided with the tape indicated that it contained 16 files, named in accordance with the VAX VMS operating system (eg BELLE.DOC). These appear in Table 2.

From inconclusive evidence obtained from the VLSI Program, it seemed likely that it was a 9-track tape, written at 1600 bits/inch, with blocks of 1600 ASCII characters. This was confirmed by inspection of a tape dump operation carried out by system staff. It further appeared that:

(a) there was a preamble file (presumably describing the contents of the tape);

(b) the file headings were standard ANSI types, which were intelligible to the IBM system;

(c) the whole set of files was repeated once.

Table 2 lists the entire contents of files.

The tape itself, with write-permit ring removed, has been named "HAYWAR", and has been retained in the possession of the author, in case further reference to it is required.

The file, HOME. is empty. Data intended for this file was supplied later, on a separate magnetic tape.

7.5 Reading a particular file

After much discussion with operators and users of the IBM system, the six-statement JCL sequence "READTAPE.CNTL", listed in Appendix I, was produced. Interpretation of this sequence can be made using reference 5.

Appendix I also gives the procedure for submitting the tape and JCL, so as to read a particular file into a catalogued dataset in the IBM system. Note that the file is written in EBCDIC characters, but is in a block format (Sub-section 7.3(f)) which cannot be viewed or printed using standard utilities.

7.6 Reformatting the file for display

It was necessary to reformat the continuous block of characters read from the tape into 72-character lines (without reference to the content of the data) so that the data can be displayed on the VDU terminal, or printed out on a line-printer. Appendix II lists the PASCAL program (and CLIST required to execute it) to read the block from one dataset and to write the reformatted data into another.

7.7 Reformatting the file into the original text

Examination of the file characters (displayed via the reformatting process of Sub-section 7.6) showed that:

(a) the data on the tape is formatted as described in Sub-section 7.3(f);

(b) the PASCAL symbols, "[", "]" and "↑", have been converted by the ASCII to EBCDIC translator to the symbols, "¢", "!" and "¬".

(c) some lines are longer than 72 characters.

(g) connection to a suitable multi-colour pen plotter;

(h) terminals available at convenient locations for prospective users;

(i) system available continuously (during working hours);

(j) adequate staff for operation, maintenance and advice.

Only two systems at DRCS seemed to approach the requirements: the central IBM 370 system; and the DEC 11 system operated by Computer Aided Processes Group, and used primarily for computer aided design, draughting and numerical machine control. Use of the latter was kindly offered by the Group Leader, who could foresee VLSI becoming a standard service offered by Engineering Workshops at DRCS.

While choice of the DEC system might have resulted in fewer problems of tape conversion (the VAX system also being produced by DEC), it fell short on all requirements (compared with the IBM system) except perhaps the availability of multi-coloured pen plotters. The IBM system was therefore chosen.

7.3 Tape conversion problems

(a) The magnetic tape does not contain the appropriate identification data for direct recognition and reading of the individual files by the IBM system, and therefore is classed as a foreign tape. Therefore a special set of JCL (job control language) had to be produced to cause the tape to be read.

(b) The ASCII character set has to be converted to EBCDIC.

(c) Some ASCII special characters can not be printed in the IBM system and must be converted to other special characters.

(d) Lower case letters, widely used in the PASCAL programs, may need to be converted to the upper case equivalent, preferred by the IBM system.

(e) Some extensions to the PASCAL language have been used in programs, and some Pragmas (instructions to the compiler) are not recognised by the IBM compiler (PASCALVS) and require conversion or modification.

(f) The files are written on the tape in large blocks of characters (with dummy characters to fill each block as required). Lines of text are written as a four digit decimal number representing the number of characters in the next line (including the four digits, themselves) followed by the line of text. No control characters, or end-of-line characters are included. It is not clear whether the IBM system can decode the text with standard utilities, so a PASCAL program was written to handle this function.

(g) A second identical version of the complete set of files follows the first one on the tape. This can be used for error correction.

(h) Tape formats, label formats and utility programs on both the DEC and IBM systems are inadequately documented and are unfamiliar to operating staff, let alone to users.

These problems seemed rather daunting, and many false starts were required. Much time of myself and others was expended in finding out how to solve them. Once the solutions were found, however, the actual task of applying them turned out to be fairly simple.

7. VAX TO IBM LIBRARY CONVERSION

7.1 General

Part of the reference material supplied by the VLSI Program to Workshop students was a magnetic (computer) tape (Sub-section 3.4(f)), containing:

(a) the PASCAL programs, BELLE, PLOTCIF, PLAGEN and GETSYMBOLS (Sub-section 3.6).

(b) documentation describing the use of (a);

(c) a CIF-code library of standard VLSI cells (symbols);

(d) documentation describing the characteristics and connection interfaces of (c).

(e) BELLE-code, CIF-code (produced by BELLE) and Symbol Library Header File (produced by GETSYMBOLS) for the standard junctions between:

(1) metal and diffusion layers (diffcut);

(2) metal and polysilicon layers (polycut);

(3) metal, diffusion and polysilicon layers (butt-contact);

(f) a file of concatenated programs, text etc and project design (in BELLE-code), generated during the Workshop, by the individual student.

The tape was produced on the VAX 11/780 by Blair Phillips, and supplied to each student at the end of the Workshop. The format of the tape was chosen by the individual student from several options, mine being a standard ANSI (American National Standards Institute) one, chosen without time for discussion with experts on the DRCS systems.

At the conclusion of the Workshop, the student ceased to have access to the VLSI Program's VAX computer, and other facilities (except for the MAIL facility). Further design work on the project had to be carried out at the normal place of employment, using computational and other facilities, as available. There was, therefore, an urgent need to read the magnetic tape, and to set to work the various programs on the local system. This section of the report discusses the difficulties encountered, and describes the methods which evolved.

7.2 Choice of host computer

The essential requirements for the host computer were seen to be:

(a) adequate size and speed;

(b) time shared terminals;

(c) efficient storage and manipulation of data sets;

(d) an efficient page-editor;

(e) properly documented PASCAL compiler;

(f) connection to magnetic-tape reader and writer;

(14) Orthogonal (90°) edge shapes are preferred.

(15) Shape rotations are restricted to 45° increments.

(16) Symbols must be defined before being called.

(17) Called symbols must be defined exactly once.

(18) Symbols must have some finite content.

(19) The CIF DD statement is not supported.

(20) No statements are allowed after a CIF End statement.

6.4 Library cell status

A CIF-code library of 40 "standard" logic cells is given in reference 2. These embrace the following functions.

(a) external connection pads of various types (10);

(b) programmable logic array components (21);

(c) inverting and non-inverting capacitance-drivers (super-buffers) including dual types (6);

(d) two-phase clock generator (1);

(e) dynamic shift register cell (1);

(f) parallel to serial data conversion cell (1).

Multi-colour plots of some of these cells are also given in the reference.

This CIF-code library was given to each student, as one file (LIBRARY.CIF) of the magnetic tape (refer to Table 2, and Sub-sections 3.3(u), 3.4(f), 3.6(g) and 7.1(c) and (d)), to be used as required, in the multi-project chip.

It was not clear whether some of these cells had been checked physically overseas. They had all been checked visually, and (in some cases) by circuit extraction program, at the VLSI Program.

The Program did not recommend the use of the following symbols:

(1) Clocklogic

(2) Padclockbar

It was strongly recommended that multi-phase clocks be generated externally to the chip, lest a failure in this logic cell should paralyse the whole project. (The "standard" clock phase-splitter relies on the deliberate use of differential time delays, inherent in the n-mos logic gates. This is regarded as bad design practice by the VLSI Program, although an alternative has not been found.)

(4) Diffusion layer connected to any pad must be spaced at least 4λ (rather than 3λ) from any non-pad connected diffusion layer area.

(5) Input pads driven by TTL level signals require buffering by a 6:1 inverter.

(6) Input pads driven by TTL level signals require a guard ring of diffusion layer connected to V_{DD}; width, 4λ; spacing from all other diffusion layer not connected to V_{DD}, at least 4λ.

(7) Pad overglass cuts must not be smaller than those in Library pads.

(8) Metal must extend at least 4λ beyond the pad overglass cut.

(9) Minimum spacing between overglass cut and:

(a) unrelated metal, is 20λ.

(b) polysilicon, is 12λ.

(c) diffusion, is 4λ.

6.3 CIF rules

The CIF rules given in Section 4.5 of reference 1, and Section 7 of reference 2, apply with the following restrictions. (Some geometrical shapes have been proscribed or constrained, and some CIF statements have been forbidden, to avoid possible bugs remaining in the CIF to MEBES conversion program, and to discourage use of procedures which have proved to be prone to human error.)

(1) Logos must be enclosed in a CIF symbol for ease of removal.

(2) Logos must obey design rules within themselves.

(3) Logos must obey design rules with respect to functional circuitry.

(4) Wires having odd λ widths must have all corner and end points on ½λ grid points. Edges then lie on whole λ grid points.

(5) Wires must have a finite positive width.

(6) Wires connect only if end points correspond (or overlap).

(7) Wires must not cross back on to own path.

(8) Boxes must have finite positive length and width.

(9) Direction vectors for boxes must have at least one non-zero component.

(10) Round flashes must have finite positive width.

(11) Polygons must have at least 3 non-colinear points.

(12) Self-intersecting polygons are illegal.

(13) Polygons must be closed (last point is same as first).

(b) Switch ON-resistance

From equation 1 - 3 of reference 1, for the non-saturated transistor:

$$R^{-1} = I_{ds}/V_{ds} = \frac{\mu\epsilon}{2D}\cdot\{\frac{W}{L}.2.(V_{gs}-V_{th})\} \quad (3)$$

Substituting actual values of F, V_{th} (for the enhancement mode transistor), V_{gs} (= V_{DD}), and assuming W = L; we find values of R of 8.5 KΩ and 11.8 KΩ for the AMI and Comdial processes; reasonably close to the nominal value of 10 KΩ /square, for the channel resistivity.

While passing currents of 53 μA and 68 μA through a switch with W = 2L (the standard gate as above) the expected voltage drop (V_{ds}) would be 0.45 and 0.80 V; not too far removed from the measured values of 0.5 and 0.6 V.

(c) Switch OFF-resistance

In the OFF condition, the current source transistor is non-saturated. Substituting appropriate values for the depletion mode transistor into equation (3) (V_{gs} = -4.55/4.2; W = L); we find values for R of 21 and 79 KΩ/square.

6. DESIGN RULES FOR AUSMPC 5/82

6.1 Dimensions

(1) λ = 2.5 μm (basic pattern length unit)

(2) Usable chip area: 2438λ x 2438λ.

(3) Maximum project size: 1200λ x 900λ.

(4) Maximum number of pads: 39.

(5) Maximum number of pads per side: 12.

(6) Pads to be spread around the perimeter for ease of bonding, and spacing of bonding wires.

6.2 N-mos design rules

The n-mos design rules of Mead and Conway (Section 2.6 and plates 2 and 3 of reference 1) apply with the following variations.

(1) The implant for depletion mode transistors is required to overlap the transistor channel area by at least 2λ (rather than 1.5λ).

(2) Any implant area must be spaced at least 2λ (rather than 1.5λ) from the channel area of any enhancement mode transistor.

(3) Buried contacts are not permitted.

5. PROCESS PARAMETERS FOR CHIP DESIGN

5.1 N-mos process parameters

Table 1 lists four sets of values of various electrical parameters pertaining to the n-mos transistors and connecting layers. The first set was taken, in the main, from reference 1, and was specified, for chip design purposes, by the MPC-coordinator.

The next two sets were an average of measurements (carried out by the MPC-coordinator) made on various test structures located on production wafers from the two Foundries. (These wafers also included the multiproject chips for AUSMPC 5/82.) The spread of the values was reported to be encouragingly small.

The fourth set was specified for design purposes subsequent to AUSMPC 5/82.

The conduction factor is given by:

$$F = \tfrac{1}{2}\mu . C_{ox} = \tfrac{1}{2}\mu\epsilon/D \qquad (1)$$

where μ = mobility of charge carriers (square metre/volt-second)

ϵ = dielectric constant of oxide layer between gate and channel (farad per metre)

D = thickness of oxide layer (metre) between gate and channel

C_{ox} = capacity per unit area of gate-channel overlap

V_{DD} = supply volts (5 V)

The capacity between the side of the diffusion layer and the substrate has been recognised as making a significant contribution to the diffusion layer capacitance. This is estimated to be 4.10^{-4} pf/μm of layer.

5.2 Derived parameters for actual wafers

(a) Gate saturation current

With the switch transistor turned hard on (non-saturated), the gate current is determined by the depletion mode, current source transistor, which is in saturation. It is given by equation 1-6 of reference 1:

$$I_{ds} = \frac{\mu\epsilon}{2D}.\frac{W}{L}(-V_{th})^2 \qquad (2)$$

Substituting actual values of conduction factor, F, and threshold voltage, V_{th}, for the depletion mode transistor, from Table 1; and assuming W = L; we find values of I_{ds} of 106 μA and 136 μA for the AMI and Comdial processes. These figures compare reasonably well with the measured values of 55 μA and 75 μA for a gate with L = 2W (the standard gate, plate 4(a) of reference 1).

4.8 Chip packaging

The 4 inch wafer set was diced into chips, and these were packaged into a 40-pin DIL by Promex. The 3 inch wafer was treated similarly by Philips, Hendon, South Australia. In both cases, pin connections were made to the pads of only one of the 5 or 6 projects on the chip, in any one package.

4.9 Chip delivery

The AMI chips (3 per project) were delivered to students on 17 August 1982. The Comdial ones (2 per project) were delivered to students on 27 August 1982.

Five unpackaged chips were also delivered: three from Comdial and two from AMI.

4.10 Photographs

One print of a photograph of both the AMI and the Comdial chips (at a magnification of about 30 times) were supplied at the same time as the chips were delivered.

4.11 MPC documentation

One copy of Designer Documentation(ref.4), one copy of the bonding map (pin connections) for the project, and one copy of the process parameters (deduced by MPC-coordinator from general test structures on a sample of chips from each wafer) were also supplied with the chips.

4.12 Chip testing

Evidence of a plan for testing the chip was one of the requirements of the report at 31 May 1982.

Students were urged to assemble and manufacture test equipment before the expected chip delivery date.

Students were given an extensive list of tips and recommendations for testing, prior to chip delivery. This particularly covered the avoidance of damage to the chip due to mechanical and electrical stresses.

Final test reports were required by 30 September 1982.

4.13 Design workshops

One-day project-design-review Workshops were held on 28 April and 10 May 1982. The student described his project and details of the VLSI design. Constructive criticism was given by other students, VLSI Program staff, and visiting consultants (Alan Bell and Carver Mead, both from USA).

I attended only the second Workshop. The pressure to meet the 3 May 1982 deadline was too great to permit the preparation required for the first.

31 May 1982 Report on circuit design

31 May 1982 CIF-code defining project

On receiving the CIF-code for a project, the MPC-Coordinator submitted it to a series of basic tests:

(1) for breach of design rules;

(2) for obvious faults such as isolated pieces of metal, polysilicon or diffusion layer;

(3) for incorrect ratio of areas of current generating (depletion mode) transistor and corresponding switching (enhancement mode) transistors in a gate.

(4) for characteristics which, though not in breach of the rules, may be expected to cause rejection of the MPC by the Foundry process.

Each designer was given the opportunity to make the appropriate corrections to the CIF design data, if necessary.

4.3 Project merging and chip allocation

Of the 85 bids for space received at Bid No 1, 60 made the final deadline at 31 May 1982. Forty-six of these were selected (by Craig Mudge, on the basis of the significance and quality of the design and the adequacy of the report) for inclusion in eight distinct dice (chips). Thus 5 or 6 projects were merged on to each chip.

Two sizes of circular silicon wafer were considered; 3 and 4 inches in diameter, respectively. A minimum of nine identical copies of each die were spread in a raster format over the 3 inch wafer, and 17 over the 4 inch wafer.

Each die (chip) is 6350 μm square at the centre of the scribe line, giving a useful area of 6096 μm square.

Merged-CIF-code-carrying magnetic tapes were forwarded to USA on 4 June 1982.

4.4 Data format conversion

The CIF-code data was converted to another format (MEBES) for use in the electron beam mask making process. This was carried out by SynMos, Mountain View, CA.

4.5 Mask writing

The electron beam writing process was carried out by Amdahl, USA.

4.6 Mask making

This was carried out by Micromask, Santa Clara, CA. There were some difficulties due to awkward polygons in the CIF-code for some Logos (project identification symbols).

4.7 Wafer fabrication

Two separate foundries were used to improve the chance of a successful MPC. One, Comdial Semiconductor Inc, Sunnyvale, CA produced a 3 inch diameter wafer. The other, AMI, Idaho, produced a 4 inch wafer.

3.8 Conduct of course

Course and guest lecturers covered all the subjects listed in Section 3.3 above in about 25, 1-hour lectures. Copious notes were supplied.

The lectures were reinforced by about 15 video tape recording lectures on the same set of subjects, being recordings of a similar course run at Stanford University, USA. Item 3.4(c) is a collection of viewgraphs displayed to students during this course.

Both lectures and video tapes closely followed the texts(ref.1,2) which cover the material in an excellent manner.

The University lecturers covered the basic theory and conducted all homework and homework tutorials. The CSIRO staff gave one specialist lecture each, on their own field of research, and conducted tutorials on project work.

Craig Mudge kept the course moving, with constant encouragement to staff and students.

Most of the theory was covered in the first 1½ weeks. The last 1½ weeks were mainly concerned with project work.

Access to the building and the computer system and other equipment was available to students, 24 hours a day, 7 days a week, and this was both necessary, and much appreciated by all students.

3.9 Projects

Students were encouraged to pair up on the main project (to be included on the multiproject chip) but few did. Most had come to the course with their own project already planned. All left with a firm start on their design, following tutorial sessions with CSIRO staff, and Ian Sutherland.

4. SCHEDULE FOR MULTIPROJECT CHIP (AUSMPC 5/82)

4.1 General

A well planned and tight schedule was laid down by the MPC-Coordinator (Rob Clarke) to ensure that each student completed his conceptual design, CIF-code design, documentation, test program and test report in time for:

(a) inclusion of the CIF-code in the MPC;

(b) inclusion of the results in the wash up sessions and final report of the MPC.

4.2 Project design phase

Deadlines not be missed for (a) were:

1 March 1982	Bid No 1 for space on MPC (name of project and estimate of chip area)
5 April 1982	Bid No 2 for space on MPC (re-estimate of size and progress of design)
3 May 1982	Floor plan and metal layout

(c) One high speed electrochemical plotter (Versatek), shared by students.

(d) Two high speed line printers were shared.

(e) A stereo microscope was permanently set up for the examination of sample chips (commercial varieties of several kinds).

(f) Photographs of various chip designs were permanently located on walls of lecture room etc.

(g) A library, slanted towards VLSI, was available.

(h) A multicolour video graphic display with digital control, calibration and coordinate display, capable of storing and displaying layout patterns at various scales and orientations, was demonstrated, but not available for student use (AED 512).

3.6 Computer programs available

(a) A DEC operating system (VMS), similar to TSO on the DRCS 370 machine, but rather easier to use.

(b) A MAIL facility for exchange of messages, data etc between groups or individuals.

(c) An efficient and simple text editor (LUDWIG) much better than that available on TSO at DRCS (but not as good as SPF).

(d) A locally generated program, in PASCAL, to generate CIF-code from coordinates fed in manually, by the student, from his multicolour layer layout diagrams (BELLE).

(e) A locally generated program, in PASCAL, to draw the CIF encoded layout on the 8-colour plotting table (PLOTCIF).

(f) A locally generated program, in PASCAL, to generate PLA layouts in CIF-code, from minterm tables, inserted manually on the keyboard (PLAGEN).

(g) CIF-code, for the standard cells listed in Section 3.3(u). These can be relocated and reoriented using BELLE and an interface program (GETSYMBOLS).

(h) A locally generated interface program, in PASCAL, to generate such essential characteristics of cells, contained in the CIF library, as are required for the calling up and manipulation of those cells by the BELLE program (GETSYMBOLS).

3.7 Program documentation

Documentation explaining the use of the programs was provided, either as computer printout, xerox diagrams, or as text via the MAIL.

The documentation varied from good to poor, although when the student difficulties became obvious, the CSIRO staff went out of their way to help. The final result was satisfactory.

It was mainly the operating system and its command procedures that were poorly documented. Copies of DEC pamphlets were available in the library, but were too detailed for the immediate requirement. They were to have been supplied for reading prior to the course, but this did not eventuate.

(q) Floor plans.

(r) Structure of VLSI circuits.

(s) Analogue circuits.

(t) Models and simulation of circuits.

(u) Standard n-mos cells.

(1) input and output pads

(2) super-buffers

(3) clock phase splitters

(4) programmable logic arrays

(5) shift registers

(v) Standard computer aided geometry specification and manipulation aids.

(1) PLA generation

(2) CIF-code specification

(3) CIF generation program (BELLE)

(4) multicolour plotting of CIF data (PLOTCIF)

(w) Wiring topology.

(x) High speed limits to synchronous circuits.

(y) Self-timed circuits.

3.4 Reference material issued to students

(a) Mead and Conway(ref.1)

(b) Hon and Sequin(ref.2)

(c) Audio-Visual Workbook(ref.3)

(d) Lecture notes.

(e) Computer program listings and documentation.

(f) Computer magnetic tape of basic programs and elements.

(g) Coloured pens etc and graph paper.

3.5 Facilities available

(a) Each student had his own video display unit and keyboard for interactive exchange of text with the resident computer system (VAX).

(b) Three 8-colour horizontal plotting tables (HP 7221C) were shared by students. One was u/s for most of the course.

3.2 Course organisation

Lectures and tutorials occupied the hours between 0900 and 1700, less an hour for lunch.

Project work, home work exercises and the gaining of experience on the computer-aided-design packages took up an extra 5 or 6 hours per day, as well as weekends.

Course leaders claimed to have compressed into a three week course, material which is usually covered in six weeks, in USA. This was only possible because of the high calibre and experience of students, and their willingness to work for long hours, daily.

3.3 Course content

(a) Familiarity with voltage logic (eg TTL) was assumed.

(b) Resistance logic (eg relay logic) was introduced and emphasised as appropriate to n-mos design.

(c) Characteristics of n-mos enhancement and depletion mode transistors, and their application to voltage and resistance logic circuits.

(d) The very significant limitations of n-mos characteristics, and means of living with these limitations.

(e) Factors affecting the speed of n-mos circuits. Calculation of time delays.

(f) The production of masks, and fabrication of n-mos silicon chips.

(g) Basic design rules for sizes and separations of the four active layers on the chip, viz:

(1) depletion mode ion implants

(2) diffusion layer

(3) polysilicon layer

(4) metalisation layer

(h) Multiphase clocking for reliable logic design.

(i) Short-term storage on gate capacitance.

(j) Static and dynamic memory, and methods of refreshment.

(k) Sequential logic circuits.

(l) Cell iteration.

(m) Stick diagrams.

(n) Scaling of process dimensions.

(o) Current and power consumption.

(p) Digital sub-system design.

(e) 1600 bits/inch;

(f) blocksize 1600 bytes;

(g) file repeated once.

8.3 Writing the tape

Again, after considerable discussion with operators and users of the IBM system, the JCL sequence "WRITAPE.CNTL", listed in Appendix V, was produced.

The appendix also gives the procedure for submitting the tape and JCL so as to write a particular file from a catalogued dataset in the IBM system on to a blank tape, in the required format.

8.4 Data transfer

The procedure, above, was used to transfer the partially completed project (in BELLE-code) to the VAX system, where subsequent design and testing was carried out by special arrangement with the VLSI Program (because of the impossibility of getting a suitable multi-colour plotter, and setting it to work with PLOTCIF, in the short time scale allowed for the AUSMPC 5/82 (see Sub-section 4.2)).

9. USING THE UTILITY PROGRAMS TO PRODUCE DESIGN CODE

The IBM system procedure for generating a complete project design in CIF-code, ready for submission to the VLSI PROGRAM for incorporation into a multi-project chip, is described in Appendix VI.

10. THE SERIAL DATA MATRIX TRANSPOSER AS AN MPC PROJECT

10.1 Binary digital communication

Modern long-distance communication links almost invariably use a guided or non-guided, single, sinusoidal, electromagnetic wave, propagating between the transmitting and receiving locations, as the carrier of the information. The strength of the carrier wave at the receiving location depends on the propagation loss, and on the power generated at the transmitting location. Other (essentially indistinguishable) low power carrier waves appear at the receiving location, due to man-made and naturally occurring electromagnetic events. These tend to mask the desired carrier, and to garble the information extracted. Communication theory is concerned with the selection of the type of modulation of the carrier wave, and the form of encoding of the information to be transmitted; to ensure an adequate rate of information, at a satisfactorily low probability of error, together with an economic use of transmitter power, signal bandwidth and channel occupation time.

One common solution (the binary digital case) uses only two distinguishable levels of carrier modulation (amplitude, phase, frequency etc) with fixed length, contiguous time-slots. The level remains constant during a slot, and may change instantaneously (ideally) at the interface time between two slots. Thus each time-slot conveys one bit of data. It is the duty of the receiver demodulator to resolve the contents of each time-slot, with the appropriate probability of success.

The slot-duration should be as long as possible (consistent with the required data rate) and the receiver bandwidth chosen accordingly, to give the best error rejection performance. If these values can not be optimised for other reasons, and the slot-duration is shorter than necessary, then the error performance can be improved by transmitting redundant information in the spare slots, using one of the many error-correcting binary digital codes. Suitable encoders and decoders operate on the desired data stream before the modulator and after the demodulator.

10.2 Burst error elimination

For a given ratio of redundant to data slots, the error correcting code can be optimised either to handle a few erroneous bits, scattered widely over the bit-stream, or a larger number of erroneous bits grouped closely together in a burst. The random distribution of errors is typical of interference by a uniform noise-like process (eg thermal noise), while the burst distribution is typical of man-made interference.

Often, both types of errors may be of significance to the system. In this case it is advantageous if some means is available for spreading out bursts of errors into widely separated individual errors. This is achieved if the bit-stream is jumbled appropriately, after the error encoding process and before carrier modulation; and if the bit-stream is reassembled in its original order, after demodulation and before error-decoding. This is the main purpose of the Serial Data Matrix Transposer.

The SDMT can be represented, logically, by a two-dimensional, rectangular matrix of single-bit storage (memory) cells; m rows and n columns. Suppose a block of m.n bits of data are entered into the memory array, in order, column by column, top to bottom, and left to right, so that the first bit is located in the top-left cell, and the last bit is located in the bottom-right cell. Suppose the block of m.n bits of data are read out of the memory array, in order, row by row, left to right, and top to bottom. After transmission, the same bits are reloaded, row by row, into a second identical memory array, and read out again, column by column.

Obviously, the overall order of bits is unchanged; but adjacent bits in any transmitted burst are separated by at least (m-1) other bits, before decoding. Providing m is greater than the processing width of the error decoder, the adjacent bits are processed separately, and appear to be randomly distributed.

It is clear from the logical operation of the store, that read-out must not begin before all the m.n bits have been written in. Therefore two stores per source will be needed at each end of the link in most applications, so that one may be writing data while the other is being read.

10.3 Multiplexing of data sources

Where several data sources are located near one end of a long-distance link, and the corresponding data sinks are located near the other end of the link, the optimum configuration usually involves a single high-speed channel carrying the total data.

Some form of time-multiplexing is used to merge the data bits at the transmitting end and to separate them at the receiving end. This usually requires a data buffer store for each source (and sink), with the ability to write data in, and read data out, under the control of two separate clocks. The outputs of all the source stores are multiplexed on to a single common data line before carrier modulation. The inputs of all the sink stores are multiplexed on to a single common data line from the carrier demodulator.

A single FIFO (first in first out) type of store, with reading and writing occurring simultaneously, is adequate for most applications. The size must be sufficient to prevent overflow due to non-uniform clocking rates at input and output.

If the time delay is not important, two separate stores could be used: one being filled while the other is being emptied. In this case, the burst error elimination transposer and multiplexing buffer can be combined into the same store. This is the function chosen for the MPC project, to be described in succeeding sections of this report.

10.4 Organisation of the transposer stores

Three possible organisations were considered for the store:

(a) a standard RAM, of total capacity not less than m.n, with no constraint on the physical arrangement of the individual cells, and with addressing via a normal binary counter, for one direction, and a modified binary counter for the other;

(b) a RAM, organised physically as a two dimensional matrix of cells with at least m rows and n columns, and with addressing via separate row and column-driving shift-registers, each register passing a single H-level along its length;

(c) a two dimensional shift-register, organised physically as a two dimensional matrix of cells, with the data bits fed in serially at one input, with the data bits fed out serially at one output, and with the direction of feed (row by row, or column by column) determined by a single control signal.

In all cases, provision is required for:

(1) selecting the input or output clock;

(2) selecting which clock moves the data row by row, and which clock column by column;

(3) remembering the state of the store, viz full, empty, partially full, or partially empty.

(4) holding the data permanently, in the absence of clock pulses.

It would also be advantageous if:

(5) the operating values of m and n could be selectable (up to some maximum value determined by the array size);

(6) both stores (with the necessary control logic) could be on the same chip.

10.5 Choice of organisation for the MPC

Implementation of the organisation type (a) is most logically made by designing VLSI control logic to drive a pair of commercial RAM chips. This results in a three-chip store which can meet the requirements, (1) to (4), perhaps (5), but not (6). It may be the most economical way to meet a real application, but does not provide a wide range of challenge to the VLSI techniques. If the RAM cells were to be included on the chip, organisation (b) provides a more elegant solution, and takes up less area of the chip. Organisation (a) was therefore rejected for the MPC.

Organisation type (b) can meet all the requirements (1) to (6), above. It is ideally suited to VLSI techniques at a commercial level. It has one major disadvantage, however. The output of each memory cell is required to drive a common bus with a very significant capacitative load. Three options are available:

(1) accept the very low speed performance;

(2) provide super-buffers at the output of each cell (or each small group of cells);

(3) use a highly-sensitive, linear, charge-to-voltage amplifier to detect the output of the selected cell.

The first option would produce a design with no real application, and would not provide a proper challenge to the designer. The second option requires far too much chip area (the buffer takes up almost half the area of the memory cell). The third option was considered far too difficult for an "amateur" designer (linear element design is not even a part of the Mead and Conway approach). Organisation (b) was therefore also rejected for the MPC.

Organisation (c) presented a wide range of challenges to the VLSI designer, without demanding non-standard techniques. It could form the basis of a real application, although it can not meet the requirement (5), above. Choice of this option for the MPC was strongly supported by Craig Mudge.

11. LOGICAL SYNTHESIS OF THE SERIAL DATA MATRIX TRANSPOSER

11.1 Two-dimensional shift-register array

Figure 4 shows a small array (3 rows, 6 columns) of shift-register cells, with data-line interconnections. Control and clocking signals are not shown. The data output of each cell connects to one data input line of the cells immediately to the right and below. A raster type of connection takes the output from cells along the bottom of the array to the input of the cell at the top of the array, and immediately to the right. A similar connection takes the output from cells along the right-hand edge of the array to the input of the cell at the left-hand edge of the array, and immediately below. The cell at the top left accepts serial data into the array through one or other of its two input lines. The cell at bottom-right provides the serial data out of the array. (Although separate X and Y outputs are shown, these always carry the same signal level). The size of the array can obviously be extended, as required, to give m rows and n columns.

Common control and clock signals are fed to each cell, so that data is shifted through the array in a TV-type of raster pattern, either vertically or horizontally, as required. When no clock pulses are present, each cell holds the data indefinitely.

11.2 Two-dimensional shift-register cell

Figure 5 shows the logical form of a suitable shift-register cell for the two-dimensional array. It consists of two inverters and four switches, which make up a static memory element with two selectable data inputs.

In the quiescent condition, the control signals, ϕ_{2A} and ϕ_{2B}, close the two switches connecting the input of each inverter to the output of the other.

The control signal, ϕ_1, ensures that both input lines are isolated from the first inverter. The cell takes the form of a static memory element, and the data at the output will be held indefinitely.

In the first stage of the shift procedure, ϕ_{2A} and ϕ_{2B} open the two inverter interconnection switches. The order of operation is not important, and both inverters maintain their previous output levels due to temporary dynamic storage at their input capacitances.

In the second stage, ϕ_1 closes one or other of the input switches, forcing the first inverter stage to take up the level of the new input data. (The selection between the two input switches is not shown in the figure.) In the third stage, ϕ_1 opens the input switch, leaving the first inverter holding the new data, dynamically, at its input capacitance.

In the fourth stage, ϕ_{2A} closes the interconnecting switch to force the second inverter to take up the new data level. This stage must be separated by a finite time- interval from the preceding one, to avoid a race condition through the array of cells.

In the fifth and final stage, ϕ_{2B} closes the feedback from the second to the first inverter, re-establishing the quiescent condition of static memory. This must occur at a finite time-interval after the preceding stage to ensure that the second inverter has attained the level of the new data before it is fed back to the first inverter.

Figure 6 shows suitable waveforms for the control signals, ϕ_1, ϕ_{2A} and ϕ_{2B}, and their relationship to the basic clock signal from which they must be derived.

11.3 Complete Matrix Transposer

Figure 7 shows the logical arrangement of the complete Serial Data Matrix Transposer, as planned for the MPC. It can carry out the full functions of burst error elimination and multiplexing of data sources, as discussed in Section 10. It can be used interchangeably at either end of the data link (ie in transmit or receive mode). Mode selection is made automatically by external connections.

$ARRAY_1$ and $ARRAY_2$ are identical m-row, n-column arrays of shit-register cells (as shown in figure 5), organised as illustrated in figure 4. The X and Y data-input lines connect to the two arrays in parallel. The single data-output line from each array, connects to the selector switch, SW_3, which determines which array feeds data to the output terminals.

Two identical Clock Splitters produce the control signals, ϕ_1, ϕ_{2A} and ϕ_{2B}, for the two arrays, from two separate clock-waves. The switch, SW_4, feeds the ϕ_1 control to either the X or Y input switch in all the cells of the array, to determine the direction of the raster-shift, and the source of the array data-input. (A nil connection to an array control line, in figure 7, is to be taken as a connection to the logical Low level.)

The switches, SW_1, SW_2 and SW_3, select the appropriate clock-wave for the two Clock Splitters from the four possible external clocks, ie X and Y raster clocks for moving data into and out of the array. Note that the Y-in and X-out clocks are used in the transmit mode; the X-in and Y-out clocks are used in the receive mode.

Two data-output terminals are supplied. One, the X and Y Data Out terminal, carries the output from whichever array is being read out from. It is used for the sink data, and for test purposes. The other, the X-data Out terminal, carries the output from the same array, but only if it is being read with an X-raster. This second output is taken via a tri-state buffer amplifier, to permit the parallel connection of several Transposers to the input bus of the multiplexed transmitting modulator.

Of the two data-input terminals, the Y-data terminal receives the source data, for transmission. The X-data terminal is connected in parallel with that of other Transposers, to receive multiplexed data from the receiving demodulator. Of course these data-input terminals will feed non-synchronised garbage into whichever array is currently being read out from, but this is of no significance as the garbage is overwritten during the next read-in phase.

Table 3 summarises the flow of data into, through and out of the Transposer, as a function of the logic levels on the three external control signal lines. It will be seen from the Table that:

(a) the Raster Select control determines whether the Transposer is in the transmit or receive mode.

(b) the Enable X-Shift control is the multiplexing gate for the data link.

(c) the Array Select control determines which array is being written into, and which is being read from.

11.4 Compliance with functional requirements

Of the six functional requirements listed in Sub-section 10.4, (1), (2), (4) and (6) have been met. Requirement (5) (selectable row and column length) is not achievable with the shift-register type of organisation (as is pointed out in Subsection 10.5). Requirement (3) (remembering the state of the stores) has not been catered for in this design, because, in a multiplexing application, the common multiplexer controller will most likely require overall control of the Transposer, including the counting in and out of data bits, and the determining of when to cease writing into an array (even if the array is not full).

Additionally, if requirement (3) were to be met (with extra logic elements) the testing of the design (as a VLSI exercise) would be made much more difficult. Furthermore, a fault in the operation of this part of the design could completely prevent testing of the clocking of the data through the arrays.

12. DESIGN OF SHIFT-REGISTER ARRAY

12.1 General considerations

Real applications of the Matrix Transposer may require shift-register arrays with a hundred or more rows and columns, but each application is likely to require a different shape and size. The VLSI technique is very suitable for this purpose, as a standard shift-register cell can be designed to use a small

chip area and low power consumption, and yet be capable of being stacked in rows and columns, as required, without the need for changes to the routing of power, signal and control connections. The actual specification of the number of cells in the rows and columns, involves only the specification of the limit of the range of the control variable in two separate PASCAL FOR-statements in the BELLE program, which call up the replication of the cells.

Arrays of this order of size, consisting of elements which must switch rapidly, and in synchronism, require clock and control signals with negligible timing delays. They can also be expected to require considerable power supply current. Thus, power and control signals need to be distributed throughout the array on metallic buses, rather than on diffusion or polysilicon rails. Since the particular VLSI process, specified for the MPC, permits only one layer of metal, these rails should run parallel to each other; otherwise the layer-crossing bridges will contribute excessive capacity and time delay. This gives a minimum of six rails; four for clocking, and two for power (figure 7).

Examination of figure 4, shows that the signal connections are short, both at input and output, but that array-length signal leads are required, both vertically and horizontally, to carry the raster-return signals. It was decided, then, that the raster-return signals would be carried by metallic rails, in the direction parallel to the power supply and clock signals; and by diffusion or polysilicon (as convenient) in the quadrature direction.

12.2 Design of the shift-register cell

The above considerations, together with the functional logic diagram of figure 5, were sufficient specification for the detailed design of the shift-register cell. The usual iterative process of trial and error resulted in a cell with the floor-plan and connections as shown in figure 8(a). Here the horizontal rails are of metal, and the vertical ones, diffusion layer. The corresponding stick diagram appears in part (b) of the same figure. (Note that here the dashed lines represent polysilicon layer, and the solid lines represent diffusion layer or metal rails. The 'I' indicates the presence of an ion-implant to produce a depletion-type n-mos transistor, used for the current source of a gate.)

The overall size of the floor-plan is 80x32λ. The replication distances are, however, 31λ horizontally, and 76λ vertically. This results in total overlap of the Gnd rails, at the top and bottom of adjacent cells; and a 1λ overlap of metal rails at left and right, and of diffusion connections at top and bottom. (The 1λ overlap was called up as a precaution because of ignorance of the details of the mask making process. All internal connected shapes are specified to overlap also.)

The two gate-switch-transistors, and the four series switches all have a channel size of 2x2λ, being the smallest size permitted under the design rules. The current source transistors have channels 16λ long by 2λ wide, giving the standard resistance ratio of 8:1 with respect to the switch-transistors (when driven by a pass transistor).

The BELLE-code, completely defining the cell, is given under the heading BDSRCELL in Appendix VII, which gives the BELLE-code for the overall MPC project.

12.3 Size of arrays for MPC

While a typically full size array would permit a full evaluation of the effect of rail length and capacity on clock delays, and also perhaps provide a usable chip for some real application, a quick estimate of the available area on a chip shared with five to eight other projects (allowing for the almost size-independent overheads of connection pads, clock-splitters and interconnection wiring) showed that this was quite impossible. A full size array would also make testing more difficult; and a faulty cell (which might prevent testing altogether) would occur with a much higher probability.

It was decided, however, that all control rails on the smaller array should be driven by super-buffers, irrespective of need. This would exercise the skills of the designer (particularly in layout), and make possible an increase in array size at a later date.

It was also decided that the complete, two-array Transposer of figure 7 should be included on the MPC, even though, in a full size system, they may be forced to reside on two separate chips. Otherwise the operation of the MPC would be rather trivial.

As a result of these decisions, an array of three rows and eighteen columns was chosen.

Figure 9 shows the final floor-plan of the project. The two arrays, with driving super-buffers below and to the right, are arranged as mirror images about the horizontal centre line of the project. (The mirror image arrangement is easily called up in the BELLE-code.)

12.4 Ancillaries to the array

For ease of testing (particularly in the event of a failure in the Data Output Selector) it was decided to bring out to pads, the two inputs and the single output of each array. These outputs could be connected back to the Data Output Selector (also with inputs and outputs brought out to pads) by means of jumpers, external to the chip. (This arrangement also eased the testing of the Data Output Selector.)

The three pads for each array are shown, in figure 9, to the left of the array. The two input pads drive the inputs of the top-left-hand cell via 8:1 isolator gates (at least 6:1 is required by item 6.2(5)) located at the immediate left of the array. These gates provide an unwanted logical inversion of the data bits fed into the array, but this is cancelled by an inverting super-buffer at the output of the array, used to drive the lowest raster-return rail which feeds the output pad, at the bottom-left of the array.

The input and output pads are standard library structures (Sub-section 6.4). A guard ring is fitted to the input pads, as required by the rule (6) of Sub-section 6.2.

13. DESIGN OF THE DATA OUTPUT SELECTOR

13.1 General

The Data Output Selector comprises the selector switch, SW_s, of figure 7, together with the tri-state and normal output pads, connected thereto, and the logic gates, EOR and AND.

As discussed in Sub-section 12.4, the two inputs to SW_5 are derived from input pads, externally jumpered to the output pads connected to the outputs from the two arrays.

The three control input signals, Raster Select (RS), Enable X-Shift (EXS) and Array Select (AS) are also derived from input pads.

The signal selection logic is carried out in a Programmable Logic Array (PLA).

The five input pads, two output pads and the PLA are shown on the floor plan of figure 9, along the bottom edge and along the lower part of the right-hand-side. This arrangement was chosen to distribute the pads along the chip circumference, and to minimise the number of cross-overs of control and signal lines, most of which are carried in the metal layer.

The input, output and tri-state output pads are standard library structures. Input pads are fitted with the required guard ring.

13.2 Design of the PLA

The PLA has five input signals: the two data outputs from the Arrays, A_1 and A_2, and the three control signals, RS, EXS and AS. Two outputs are required: the data output, DO (feeding the two output pads) and the tri-state control signal, TS.

Examination of figure 7 shows that the logical operation of the PLA is defined by:

$$\left.\begin{aligned} DO &= A_1 . (RS.\overline{AS} + \overline{RS}.AS) + A_2 . (\overline{RS}.\overline{AS} + RS.AS) \\ TS &= \overline{RS}.EXS \end{aligned}\right\} \qquad (4)$$

Five logical min-terms appear in equation (4). The required input data file for the program PLAGEN is therefore:

```
5 5 2
xx01x 10
1x0x1 01
1x1x0 01
x10x0 01
x11x1 01
```

where the columns apply to A_1, A_2, RS, EXS, AS, TS and DO respectively.

Note that the first line specifies the number of inputs (5), the number of min-terms (5) and the number of outputs (2). The following lines define each min-term in terms of the contribution of each input (x: don't care; 1: direct effect; 0: inverted effect), and the output it affects (1: affects output; 0: does not affect output).

The CIF-code, generated by PLAGEN, was added to the CIF-library of standard cells, under the name PLA500.

13.3 Ancillaries to the PLA

A gate ratio of at least 6:1 (rule (5) of Sub-section 6.2) is required at each input of the PLA, because each input is derived from an input pad. The 4:1 gate at each input, generated by PLAGEN, was easily increased to 8:1, by

doubling the width of the channel of the pull-down transistor, using geometry defined under the heading PLAPROJ in the BELLE-code of Appendix VII.

14. DESIGN OF THE CLOCK SELECTOR SWITCH

14.1 General

The Clock Selector Switch comprises the switches, SW_1, SW_2 and SW_3 of figure 7.

The function could be carried out by a PLA, with 7 inputs, 2 outputs and 8 minterms, and taking up a chip area of about 164x122λ. It could also be carried out by a pass-transistor network (enhancement transistors used as series switches in relay logic type networks) with no more than three pass transistors in series (the maximum number recommended by Mead and Conway). The latter realisation was chosen for the additional design experience, and to allow evaluation of the speed of response of this type of cell. It took up about 132x83λ of chip area.

For ease of fault finding, and for ease of testing the speed of response of the pass-transistor network, the output to the Clock Splitter of $ARRAY_2$ is fed via standard library output and input pads and an external jumper wire.

The three control input signals are in parallel with the control inputs to the Data Output Selector PLA.

The four Clock-input signals are derived from standard library input pads.

All input pads are fitted with the required guard ring.

The five input pads, the single output pad, and the Clock Selector Switch cell are shown on the floor plan of figure 9, along the top edge and along the upper part of the right-hand-side. Again, the arrangement of pads and logic cell was chosen to distribute the pads along the circumference of the chip, and to minimise the number of cross-overs of the control and signal wires feeding the cell.

A pair of non-inverting super-buffers take their input signals from the Clock Selector Switch (via the output and input pads for the second Clock Splitter) and drive the relatively long connection lines to the two Clock Splitters. These buffers are shown on figure 9 at the bottom right corner of the Clock Selector Switch and adjacent to the input pad.

14.2 Design of the pass-transistor network

Figure 10 shows the stick diagram of the Clock Selector Switch, less the two super-buffers. Dotted lines represent polysilicon layer, and solid lines represent diffusion layer (except for the V_{DD} and Gnd rails, and the cross-over loops, which are in metal). The operation is a direct analogue of the switch arrangement shown in figure 7.

Twelve depletion layer transistors, along the top of the diagram, are current sources for the twelve inverting gates used. These are identical transistors with a channel size of 8x2λ. The first four, and last two, gates have a ratio of 8:1, required to buffer the TTL levels from the input clock signals, and the output of the pass-transistor logic network. The double logic inversion therefore cancels out, but the in-between section operates in inverted logic. (The only effect of this logic inversion is that the low logic level input to SW_3 becomes a high logic level input, ie V_{DD}).

The other six inverting gates provide direct and inverted versions of the three control signals, RS, EXS and AS. Again, 8:1 ratios are used for three of them, to buffer the TTL levels from the input pads. The other three use the normal 4:1 ratio, being fed from normal gates.

All nine pass-transistors have the minimum channel size of 2x2λ.

The BELLE-code for the network appears under the heading CLOCKSEL in Appendix VII.

15. DESIGN OF THE CLOCK SPLITTER

15.1 General

Figure 6 shows the required waveforms of the three outputs of the Clock Splitter, ϕ_1, ϕ_{2A} and ϕ_{2B}, with respect to each other and to the selected clock input waveform. Although a transition occurs in each of the three outputs for each transition of the input, the quasi-simultaneous output transitions must, in fact, occur in a fixed order, with a finite delay between them. Otherwise, race conditions may occur in the Two-Dimensional Shift-Register Array, as discussed in Sub-section 11.2. These finite delays cannot be derived digitally from the input clock, because they are much shorter than half a clock period. Thus the delays must be produced by analogue means.

Just this problem of ordering transition delays has already been faced by the designer of the standard library cell called PADCLOCKBAR. This cell produces two output clock waves, one (ϕ_1) in-phase, the other (ϕ_2) out of phase, with the input clock; and with the high period of ϕ_1 beginning earlier, and ending later, than the low period of ϕ_2. An examination of this design shows that it comprises two OR-gates, two inverting super-buffers and three inverting gates, connected as shown in figure 11(a), or the re-arranged version of figure 11(b). The operation depends on the fact that the OR-gates act as OR-gates for rising transitions at the subsequent ϕ output (and therefore the transition in ϕ follows the transition in the input clock), but act as AND-gates for the falling transitions of the subsequent ϕ output (and therefore the transition in ϕ follows the transition in the other ϕ output, delayed by two gates and a super-buffer).

The VLSI Program staff advised students not to use this type of circuit in projects (Sub-section 6.4). Opposition was based on:

(a) analogue delay generation is bad circuit practice

(b) such circuits require extreme care in design

(c) the library cell PADCLOCKBAR had not been evaluated experimentally.

The alternative was to generate the three clock waveforms, externally to the chip, using TTL circuit elements. In this particular application, this is most unsatisfactory. Many extra pad connections would be required, and the TTL circuits would have a large effect in determining the overall speed of the Transposer, Besides, if VLSI is to have a real application in the world of digital logic, this problem has to be faced and overcome. All commercial logic families use these analogue techniques (although many problems were experienced with early versions of the designs).

It was decided, then, that the risk would be accepted and the Clock Splitter would be located on the chip. The penalty of failure would be the likely inability to check out the operation of the Shift-Register Array, and a virtually wasted project.

15.2 Clock splitter design

Figure 12(a) shows the logic diagram of a clock splitter, based on the operation of PADCLOCKBAR, but which produces the three clockwaves, ϕ_1, ϕ_{2A} and ϕ_{2B} of figure 6. It comprises three AND-gates, SB_1, SB_2 and SB_3 (which provide the three clockwaves) and three delay networks, DL_1, DL_2 and DL_3. Similar reasoning shows that a positive going transition of the input causes ϕ_{2A} and ϕ_{2B} to undergo coincident negative going transitions (delayed only by SB_2 and SB_3, respectively). The corresponding positive going transition of ϕ_1 is further delayed by DL_3 and SB_1. A negative going transition of the input causes ϕ_1 to undergo a negative going transition (delayed only by SB_1). The corresponding positive going transition of ϕ_{2A} is delayed further by DL_1 and SB_2. That of ϕ_{2B} is delayed further again by DL_2 and SB_3. The greater the delays in the networks, the larger is the safety margin against races in the Shift-Register.

Figure 12(b) shows the logic diagram of the circuit chosen for the actual Clock Splitter. Duplication of the first gate and delay network, plus the addition of the gates, OR and A_3, allows the X-Y direction selection function (SW_4 of figure 7) to be included. Gates, A_1 and A_4 are needed to correct the logic when NOR-gates, ISB_1, ISB_{2A} and ISB_{2B} are used. DL_2 can substitute for DL_3, in figure 12(a), if the connection to the first AND-gate is taken from ϕ_{2A}, instead of from ϕ_{2B}.

Super-buffers were considered essential to drive the clock rails throughout each Shift-Register Array. Examination of the structure of the Inverting Super-Buffer showed that one or two extra pull-down transistors could be added in parallel with the existing one at the input. With this addition, it becomes a two or three-input NOR-gate; appropriate for ISB_1, ISB_{2A} and ISB_{2B}.

15.3 Clock skew and delay elements

Ideally, the generation and distribution of clock waves to an array of identical elements should be such that:

(a) the clock transitions arrive at the cells with sufficiently small rise time

(b) the transitions of different clock phases should arrive at each cell of the array in the correct time sequence, and with adequate spacing

(c) each transition of any one clock phase should reach every cell of the array simultaneously

to avoid the incorrect transfer of information within an element, and between adjacent elements.

With the master-slave type of operation chosen for the cells of the Transposer Array, item (c) is not of critical importance, although gross clock-skew over the array may limit the maximum clocking frequency possible. Item (a) is covered by the use of rail-driving super-buffers, and metallic layer horizontal clock rails through the Array (to minimise the series resistance and shunt capacitance). The vertical tie-rails (at one or both sides of the Array)

have been chosen to reside in the diffusion layer, however, and could cause excessive degradation, particularly if the number of rows of cells is increased. Item (b) is covered by the appropriate staggering of transitions at the Clock-Splitter, and by ensuring that this is not negated by poor routing of the signals between the Splitter and the cells.

At first, the location of the ϕ_1 and ϕ_{2B} drivers on one side of the Array, and the ϕ_{2A} driver on the other, looked attractive. Closer examination showed that the apparent advantages (of using the horizontal array rails as delay elements) did not materialise, and the disadvantages (much greater chip area used, and need for a clock input rail through or round the Array) were large.

A much better scheme, and the one chosen, is illustrated in figure 13. All rail drivers are located on the right-hand-side, with those for ϕ_1 at the bottom of the array, and those for ϕ_{2A} and ϕ_{2B} at the top. The diffusion layer vertical tie-busses, loaded with the horizontal array rails, act as delay lines, DL_1 and DL_2, and compensate for all the distribution delays. An inverting super-buffer, below the ϕ_1 drivers, is used for the inverting gate, A_1, in the input clock signal.

15.4 Clock-Splitter BELLE-Code

The super-buffers, ISB_{2A} and ISB_{2B}, together with the logic gates, OR, A_3 and A_4 of figure 12(b), are defined under the heading SBMOD1 in the BELLE-code of Appendix VII.

The two super-buffers, ISB_1, are defined under the heading SBMOD2.

The X-Select Control gate, A_2, is defined under the heading SB2IF. This contains two inverters in cascade, and provides an 8:1 buffer for the TTL control signal, AS. It is clear from figure 7, that the gate, A_2, must feed the Y-super-buffer in the case of the second Array. This is achieved by fitting SB2IF to the second Array without mirror inversion. The second version of SBMOD2 is mirror inverted.

The clock inverting gate, A_1, and the interconnection rails of figure 13, are included in the BELLE-code for the Array, under the heading SRARRAY. This element also calls up SBMOD1 and SBMOD2, and locates them appropriately with respect to the Array.

16. PROJECT-DEFINING BELLE-CODE

16.1 General

Appendix VII gives the complete BELLE-code for the Serial Data Matrix Transposer, from which the final CIF-code was generated (as described in Appendix VI) for submission to the MPC Coordinator at the VLSI Program. Only the LOGO-code (which placed the identification label on the chip) and the node-labels (which are used by circuit extraction programs (Sub-section 4.2)) have been omitted.

16.2 Library Cells

Standard library cells, defined elsewhere in CIF-code, and called up by the BELLE-code, are listed below.

(a) PADIN

(b) PADOUT

(c) PADTRISTATE

(d) PADVDD

(e) PADGROUND

(f) INVERTINGSB

(g) NONINVERTINGSB

(h) INVERTINGSBPAIR

(i) NONINVERTINGSBPAIR

e only non-standard cell, defined in CIF-code, and included with the above, PLA500 (Sub-section 13.2).

.3 Element flow sequence

16.3.1 Minor elements defined, include:

(a) PULL4 is a normal 8x2λ depletion mode pull-up transistor without the 2λ wide diffusion layer channel strip. It is used in conjunction with a 2x2λ or 2x4λ pull-down transistor to produce a 4:1 or 8:1 ratio inverter gate.

(b) PADINM is a standard PADIN, surrounded with a 4λ wide diffusion layer guard ring connected to V_{DD}.

(c) CLOCKSB is a standard NONINVERTINGSBPAIR with input and output extensions and additional diffusion layer needed to give an 8:1 input ratio gate necessary for the signal from the input pad.

16.3.2 More significant elements, already defined, include:

(a) BDSRCELL (Sub-section 12.2)

(b) PLAPROJ (Sub-section 13.3)

(c) CLOCKSEL (Sub-section 14.2)

(d) SBMOD1 (Sub-section 15.4)

(e) SBMOD2 (Sub-section 15.4)

(f) SB2IF (Sub-section 15.4)

16.3.3 SRARRAY is one array of three rows and 18 columns of BDSRCELLs; with connections and rails along the top and sides; with data input buffer gates at the top left; with raster-return super-buffers along the bottom and right-hand-side; with SBMOD1 and SBMOD2 at the right-hand-side; and with pads along the top (including V_{DD}) and left-hand-side (but not V_{GND}).

APPENDIX I

READING FILE FROM VAX TAPE INTO IBM DATASET

JCL sequence "READTAPE.CNTL"

```
//JWHZ JOB ,,CLASS=1,MSGCLASS=T,NOTIFY=JWH
//       EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=TAPE1600,DISP=OLD,
//          VOL=SER=HAYWAR,
//          DSN='PLAGEN.PAS',
//          LABEL=(26,AL),
//          DCB=(DEN=3,BLKSIZE=2048,LRECL=2048,RECFM=FB,OPTCD=Q)
//SYSUT2 DD DISP=(,CATLG),UNIT=SYSDA,DSN=JWH.PLAGEN.BLK,
//          SPACE=(TRK,(20,15))
//SYSIN DD DUMMY
```

Notes on parameters

1) JWH is the user's user-identification.

2) IEBGENER is the IBM utility program which reads the tape.

3) SYSPRINT is the DDname of the message dataset required by IEBGENER.

4) SYSUT1 is the DDname of the magnetic tape, being the input dataset equired by IEBGENER.

5) TAPE 1600 defines the device (tape drive operating at 1600 bits/inch).

6) HAYWAR is the name written on the reel of magnetic tape, identifying t to the system operator.

7) PLAGEN.PAS is the name of the required file, contained in the label ritten on the tape.

8) 26 is the sequence number of the particular file "PLAGEN.PAS" as ritten on the tape.

9) AL specifies that the tape has ANSI labels.

0) DEN=3 is a sub-parameter specifying 1600 bits/inch, 9-tracks.

1) 2048 is the maximum length of ASCII records on magnetic tape.

2) OPTCD=Q specifies that translation from ASCII to EBCDIC is required.

3) SYSUT2 is the DDname of the output dataset required by IEBGENER.

4) , CTLG specifies a new dataset that is to be kept in the user's talogue.

5) JWH.PLAGEN.BLK is the name of the dataset to contain the file, read om the tape (in block format).

REFERENCES

No.	Author	Title
1	Mead, C. and Conway, L.	"Introduction to VLSI Systems". Addison and Wesley, 1980
2	Hon, R.W. and Sequin, C.H.	"A Guide to LSI Implementation". 2nd Ed., Xerox Palo Alto Research Center Report, 1980
3	Fairbairn, D., Mathews, R. and Newkirk, J.	"Introduction to VLSI Systems Design". VLSI Technology Inc, 1982, (Audio- Visual Course Workbook)
4	Clarke, R.J.	"AUSMPC 5/82 Designer Documentation". CSIRO VLSI Program, 1 August 1982
5	-	"OS/VS2 MVS JCL Release 3.8". IBM Document GC28-0692-4, May 1979

Section 18. It is possible that the measured limit was in fact the limit of the Test Clock (which gave a very poor square wave at 8 MHz) or that of the standard TTL logic chips generating test signals for the chip under test.

(3) Satisfactory operation over a supply voltage range of 3.5 to at least 6.0 V, and with a corresponding change in current consumption of about 4 mA (compared with a mean consumption of 19 mA at the nominal supply voltage of 5 V) shows the broad tolerance to supply voltage fluctuations obtainable with n-mos circuits.

(4) The supply current, under nominal test conditions of +5 V supply, 500 kHz clock frequency, transposition of a pseudorandom data sequence, and data output from the tristate output pad, averaged 18.1 mA (AMI chips) and 22.9 mA (for the only usable Comdial chip). This is in good agreement with the predicted consumption of 17.7 ± 7.6 mA as estimated in Section 17.

20. CONCLUSION

lead and Conway method of design provides a good introduction for the ·ation of workable, moderate complexity, moderate speed, NMOS, VLSI circuit ;.

:IF format of circuit definition is an excellent interface between the .cations designer and the silicon processor.

the VLSI experience already obtained, and with the computer aided design ıges now set to work on the DRCS IBM main frame computer, it should be .ble to design another project, of similar complexity, in 1 to 2 months. ıis is equivalent to several printed circuit boards of medium scale integ- ın digital logic, it seems to be an economic way of producing Defence :ronics.

: from the VLSI experience gained, the project has indirectly provided an ıous amount of experience in Computer Operating Systems (IBM and VAX), and ıe use of the PASCAL Programming language.

/hole VLSI Workshop and Project has been an interesting, useful and mind- :ching process, culminating in the development of a fully working circuit, ı met all requirements and expectations.

21. ACKNOWLEDGEMENT

ı a pleasure to acknowledge the great assistance of many people without whom project would never have been brought to a successful conclusion; in .cular:

ı) Z. Jeffrey, M. Raymond, D. Gapp and S. Matulich, for suggesting the ıtrix Transposer as a suitable project, and for defining many of its :quirements.

›) Craig Mudge and the VLSI Program Team, for much patience and practical ısistance in the manipulation of CLISTs and the use of design tools, and for ıe availability of the VAX System during the design phase of the project.

:) K. Sarkies, for solving the problems of exchanging data sets on magnetic ıpe, between the VAX and IBM Computer Systems.

(4) Dual Channel Oscilloscope (10 MΩ, 14 pf, 100 MHz)
Tektronix Type 454 with Type P6063 Probes.

(5) Current Meter (100 mA, DC)
Avometer Type 8 Mk 3.

(6) Leitz Universal Wide Field Research Microscope.

19.5 Chips under test

Five chips were available for testing. Three were fabricated on 4 inch wafers by AMI, Idaho, USA, and packaged by Syn Mos. These have a greenish tinge under the microscope. They were numbered, 1, 2 and 3. Two were fabricated on 3 inch wafers by Comdial, Sunnyvale, USA and packaged by Philips, Adelaide. These have a bluish tinge, and a sharper pattern. They were numbered, 4 and 5. All were fabricated from masks made by Micro Mask, Santa Clara, USA.

19.6 Test results

(1) Chips 1, 2, 3 and 5 operated as expected at clock speeds up to 3 to 4 MHz (the application calls for a maximum clock speed of 150 kHz). They were rated as 100% successful.

(2) Chip 4 returned data almost completely unrelated to the input data. The nature of the output was such as to permit a fairly confident assessment of the location and nature of the faults. It appears that the seventh column of shift-register cells in Array No 1 has a cell stuck at Low (in row 1) and at High (in row 2 from the top). The sixth column of Array No 2 appears to have cells stuck at High (top row) and Low (middle row). This suggested a linear fault line running almost vertically for about 1200 μm. No evidence of this was visible using the microscope in the interference-contrast mode, and with a magnification of 400. Presumably the fault must lie in the silicon substrate. The remainder of the chip operated as expected, at the same maximum clock rate as the others.

(3) The supply voltage on the chip was reduced until data transposition collapsed. This occurred at voltages of 2.7, 2.8, 2.9, 3.4 and 3.3 respectively. Chip supply currents were measured at this voltage, at +4, +5, +6 V, and under other conditions as specified in Table 14. Results are shown in that Table.

19.7 Discussions of results:

(1) The complete success of the clocking of the two dimensional shift register arrays goes some way towards justifying the gamble in choosing wiring and gate delays to provide the race-avoiding time delay between otherwise simultaneous transitions of the three clock signals. (The same technique is used in the standard Caltech Clock Logic Cell, the use of which was not recommended for use in the AUSMPC.) For the circuit arrangement chosen, calculations show that the required time delays are absolutely positive (although small) irrespective of the magnitude of individual element delays.

(2) The measured clock frequency limit of 3 to 4 MHz gives a very satisfactory margin over the maximum intended frequency during use (150 kHz). It is somewhat less than the value of 5 MHz predicted in

The switch, S_{10}, permits the inhibition of the pseudorandom data sequence generator.

Standard SN74 series, TTL chips are used in the realisation of the Test Assembly. All switches, except S_1 and S_2, are buffered by TTL gates (or set-reset flip-flops) from the chip under test, to give realistic TTL level inputs under static conditions.

Test points 6, 7 and 10 on the hardware, are buffered by inverting gates. They show inverted signals compared with figure 26.

Note that TP_4 and TP_5 carry inverted data levels from the two Arrays, and these are also fed into the Output Selector. Consequently TP_1 and TP_2 carry data in the non-inverted form. TP_3 also carries inverted Clock Out No 2 signals.

All switches in figure 25 are shown in the position with the manually operated toggle upwards.

19.3 Functional test procedure:

(1) The chip under test is fitted into the Test Assembly; the 5 V supply is applied; and the Test Clock In signal is applied (TTL levels, at a frequency of 80 kHz).

(2) The oscilloscope is triggered from TP_7, and adjusted to show two consecutive periods of the 54-bit data sequence.

(3) The switches are set as shown in Table 13, where High, Low and Middle refer to the position of the manually operated toggle. Signals expected at various test points are specified in the Table.

(4) Where an X is specified for the toggle position, all combinations are to be tried, with no changes expected in output.

(5) If the signals are not as specified, the fault is located using the manually operated switches and test points in an adhoc manner.

(6) The chip supply current is measured under dynamic data conditions, as for the first line of Table 13.

(7) The frequency of the Test Clock In is raised until the dynamic data transfer collapses. The maximum chip clock frequency is half of this frequency.

(8) Total chip supply current is measured for a variety of supply voltages, clock frequencies and switch settings.

19.4 Test equipment used

(1) Special Purpose Functional Test Assembly (figure 25).

(2) Power Supply (5 V, 100 mA)
Advance Type PP3.

(3) Squarewave Generator (TTL levels, 0 to 10 MHz)
Datapulse Type 110B.

Figure 24 shows the pad connections to the pins of the 40-pin Dual-In-Line package, which contains and protects the delivered chip. Sufficient internal logic is also shown to identify the location of inverting gates and super-buffers.

19.2 Functional Test Assembly

A special purpose Functional Test Assembly was constructed to hold the chip package (by means of a zero-insertion-force socket) and to provide the necessary power supplies, input data, clock signals, control signals and output signal buffer amplifiers. It is capable of completely evaluating the entire chip circuit. A 54 bit, pseudo-random serial data stream is used to verify correct data transfer from cell to cell, and to verify that proper matrix-transposition occurs between the input and output.

The logical operation of the unit is as shown in figure 25. An 8-stage shift register, SR, generates a 54-bit, serial pseudo-random sequence at the output of its eighth stage (TP6). Feedback via the exclusive-OR gate, EO_1, would generate a 63-bit maximum-length-sequence. The feedback from A_1 causes 9 bits of the sequence to be bypassed, reducing the length to 54 bits (see the waveforms of figure 26). The gate, A_2, has a dual function. Feedback via OR_1 ensures that the sequence cannot latch up with SR containing all lows. Secondly, it allows the flip-flop, FF_3 to change state (via EO_2) after each 54-bit period. The output of FF_3 (via TP_7 and S_3) controls the Array Select line, causing the two shift-register-arrays on the chip, to interchange roles.

FF_3 and SR are clocked by the high-going transition of the output of the frequency-dividing flip-flop, FF_1, driven by the square wave Test Clock In.

The four separate Clockwave inputs to the chip are selected from the flip-flop, FF_2, via the switches S_4 to S_7. The output of FF_2 follows that of FF_1, delayed by one half cycle of the external clock (or ¼-cycle of the chip clockwave). This ensures that the data signal at TP6, is latched for the full positive half cycle of the clockwave from FF_2, thereby avoiding race conditions during the loading of the first cell of the Arrays. (Note that for ease of representation on figure 26, the ¼-cycle offset between FF_1 and FF_2 has been suppressed in the lower seven waveforms.)

Waveforms in the lower half of figure 26 show the master and slave parts of the first cell of an Array, and the contents of the slave (output) part of the last (54th) cell for the three conditions:

(a) read out under control of the Y-clock, assuming the read in (during the previous period) was also under control of the Y-clock. (Read in and out under the control of the X-clock would give the same results.)

(b) read out under control of the X-clock, assuming the read in (previous period) was under control of the Y-clock.

(c) as for (b) but with clocks interchanged.

The Snapshot Ladder defines the contents of the 54 slave-cells at one instant of time. The cell numbering depends on whether the X or Y clock is being considered. Only the 1st (top left) and 54th (bottom right) cells are numbered the same in both considerations.

The switches, S_3 to S_9, permit the selection of fixed logic levels as an alternative to the dynamic signals discussed above. S_1 and S_2 provide only fixed logic levels for the Raster Select (Transmit/Receive) and Enable X-shift control signals.

18.6 Output PLA

The equivalent circuit of the Output PLA (which selects the output data from the two Arrays) is shown in figure 23(b), together with the half voltage output delays. Although this type of selector switch also has rather long switch rails, the capacity is a little smaller (due to mostly metallic rails, rather than diffusion layer ones), and the transistor sizes are larger, resulting in faster operation. The even number of inverter gates, used in this particular application, tends to equalise the two values of delay, and hence to reduce the pulse distortion.

18.7 Summary

The estimation of the operating speed of the chip circuits suggests that the whole system should operate satisfactorily to beyond 5 MHz in clock frequency. This is well in excess of the envisaged requirement of 0.15 MHz.

The practical performance (Section 19) conformed quite closely to this estimate.

Further simulation would be required to determine the sensitivity of the operation of the array cell to the ratios of its node capacitances.

It is very clear that circuits of this type, which use charge storage on circuit stray capacitances, require very careful examination. Simulation plays an important role in the understanding of the operation, because the gate current relationships are extremely non-linear over the range of gate and output voltages encountered.

19. TESTING THE CHIP

19.1 Built-in test features

As it is virtually impractical to insert or extract test signals at nodes within the chip layout, it is necessary that appropriate test nodes are brought out to extra input and output pads, when the layout is designed. To reduce the number of extra pads, it is often recommended that test nodes be sampled, in parallel, into a test shift register. The readout can then be in a serial mode, via a single output pad. Alternatively, signal paths can be broken by means of an output and input pad-pair to permit both signal monitoring, and signal insertion.

The Serial Data Matrix Transposer, however, is essentially two large serial shift registers, together with the necessary clock phase splitters, and the selector switches for clocks and data streams. Furthermore, the sensitivity of the timing of the operation of the Arrays and the Clock Splitters to the circuit capacitances, and the number of separate clock signals into each Array (four), make it rather impractical to break many of the critical signal paths: this would invalidate all the operating speed estimations.

Therefore the gamble was taken to restrict test nodes to break-pad-pairs isolating only the Output Selector from the Array outputs, the Clockwave Selector from the second Clock Splitter, and the X and Y-data inputs to the two Arrays. If the Clock Splitter or Array had not worked, there would have been little hope of finding out the reason.

The simulation was repeated with the output capacitor increased to 0.36 pF in the centrally located non-broken cell. To help simulate more cells ahead of the modified one, the initial conditions for the source half cell were varied at the beginning of each cycle, in accordance with the transient results from Table 11. Table 12, and the dotted curves in figures 21 and 22, show the results. The only significant effects were in the second half of the clock cycle, for the modified cell (as shown on the right of the figures); and in the first half of the clock cycle, for the subsequent, unmodified cell (as shown on the left of the figures). The effect is very significant for the Low to High step function (of figure 21) where the value of V_1 drops to 2 V for a few nanoseconds after the clock wave ϕ_{2_b} goes High.

At first sight this looks capable of preventing the correct data transfer operation, but, due to the delay in any corresponding change in the level of V_2, there is still plenty of margin in hand. It is clear, however, that extra output capacity is destabilising and should be avoided. The use of super-buffers as row and column drivers was obviously well chosen.

(e) Other input sequences

The results of the step function simulation showed that cell transients at the end of a cycle are independent of the transient condition of the driving cell, and have largely reached steady state conditions after two cycles. It follows that the end of cycle voltages for a ring of four cells, started with levels 0110 or 0101, are predictable from the step function results, and this proved to be the case. Within-cycle transient conditions are less onerous than those for the step function. Therefore figures 21 and 22 cover the worst possible cases.

18.5 Clock selector switch

Figure 23(a) shows the equivalent circuit, derived from the actual chip areas, and the half voltage output delays for the two directions of signal change, derived from the same type of simulation.

The capacities of the switch rails are rather high, and, together with the well known high resistance of the series switch transistors, cause a relatively long delay, particularly in the negative going direction (when the series switches are taken High). This causes a delay distortion of 20.8 ns, which is in the same (undesirable) direction as that of the non-inverting super-buffers (Sub-section 18.3(e)).

The effect of this extra distortion on the operation of the shift-register array, was examined by simulation. Surprisingly, although the duration of the ϕ_{2_b} clock wave is reduced to only 6.9 ns, the data step function passed correctly through the cascade of cells (including one with an output capacitance of 0.36 pF). For the transition from High to Low, there were only minor changes to the node voltages at the end of a cycle. For the inverse transition, the changes were quite significant at the input and output nodes (1.99 and 2.72 V, compared with 2.47 and 3.23 V for the earlier simulation). At first sight, it appears that the low input voltages would cause the input stage of the cell to reverse its state. In fact it does not persist long enough for this to happen, before the next cycle begins. There is, of course, no problem during the closed loop state, because it is only at the end of this (foreshortened) state that the clock distortion begins to take effect. This all only confirms the sheer robustness of the cell operation.

(c) Simulation details

A computer program was written (based on the previously discussed models) to simulate four cells (each as shown in figure 20) connected in a ring. Three cells always had the lower value of output capacity, and represented main array cells. The fourth cell could take either value of output capacity. The time increment was again taken as 0.1 ns.

One cycle of simulation covered one cycle of the clock wave, with the switch timing based on the data of Table 10, modified to include the delay unbalance in the non-inverting super-buffer driving the Clock Splitter. Zero time corresponded with the Low to High transition of the ϕ_1 wave. End time was one clock period later.

The clock period was taken as 200 ns. This was a convenient round number, a little larger than the minimum value indicated by the second column of data in Table 10, but a little smaller than the experimentally determined minimum period for operation of the VLSI chip.

(d) Data step function

Perhaps the most simple sequence of data bits through a cell is the step function. A long succession of low level data bits is followed by a long succession of high level data bits (or vice versa). If the length of each succession is great enough, all transients will die out and the cell voltages will all take their final steady state value.

This case was simulated with only four cells, by effectively breaking one in its middle. The input section then loads the last of the non-broken cells, and the output section feeds the first one. Initially, the three unbroken cells have the four capacitor voltages set to the steady state values, corresponding to a stored Low. The two capacitor voltages at the input of the broken cell are set likewise. The two capacitor voltages at the output of the broken cell are set to the steady state values, corresponding to a stored High.

After three complete cycles (at the beginning of each of which, the capacitor voltages of the split cell are reset to their original initial condition) we have a good set of transient performances corresponding to 1, 2 and 3 cycles of transient decay in a cell, when driven from another cell in various stages of transient decay. Obviously, with four cells, the length of the bit succession is rather small. For the clock period chosen, however, it turns out to be adequate. Table 11 gives the voltages at the beginning and end of each of the three cycles, for both directions of step. It is clear that the transient is virtually over by the end of the second cycle, and that the operation of one cell is largely independent of the transient state of the driving cell.

A little thought shows that the most difficult transient response under data step function excitation, occurs for a cell at the steady state condition, driven via a chain of one or more other cells. The third unbroken cell meets this requirement in the third cycle of simulation, and the solid lines of figures 21 and 22 illustrate the progress of the transients during this cycle, for the two directions of step. In both cases, the transients generated during the first half clock cycle, die out completely within the cycle. It is only the High going voltages in the second half clock cycle which exhibit unfinished transients.

(c) Clock rail driver delays

Figure 17, parts (c) and (d) show the circuits, simulated separately, to represent the remainder of the logic elements in figure 12(b). Again, the results are given in figure 18.

(d) Estimated time sequence

Table 10 gives the time sequence, deduced from the time delays of figure 18, for both directions of clock transition. Dead times of 20.2 and 36.4 ns occur between the negative going transition of one clock wave output and the positive going transition of the other.

(e) Unbalanced delay of non-inverting super-buffer

Figure 19 shows the equivalent circuit and the half voltage output voltage delays (with step function excitation) for the non-inverting super-buffers used to drive the clock input rails to the Clock Splitters. The output capacity of 0.90 pF refers to the second Array. That of the first Array would be somewhat smaller.

Note that the time delay unbalance of 5.2 ns should be added to the figures in the second column of Table 10, to get a more accurate picture of the operation of the Clock Splitter.

18.4 Shift register array

(a) General

Each shift register cell consists of two inverting gates, each with input and output capacity capable of storing charge. Transistor switches interconnect these capacitances in a sequence designed to transfer voltage levels (and hence logic levels) on a master-slave basis, as illustrated in figures 4, 5 and 6.

Data fails to transfer satisfactorily if, at the end of any one stage of the sequence, the voltages stored on the four capacitances are inappropriate for the next stage of the sequence. This can happen for two reasons:

(i) if insufficient time is allowed for a stage of the transfer, the stored voltages may differ excessively from the intended assymptotic values.

(ii) if a closed feedback loop exists (as in the case of the indefinite storage stage) and if the gate capacitances are ill proportioned, transient voltage levels may cause the loop to trip into the wrong state.

Both of these possibilities are examined for the case of typical circuit parameters.

(b) Cell characteristics

Because of the photographic reproduction, all cells in an array may be assumed to have the same characteristics of gain, capacitance, etc, except for cells at the bottom and right-hand edges of the array (which have an increased output capacity due to longer wires and to the inputs of super-buffers). An estimate of these capacities, based on the areas of wires and the data of Table 1, has given the results shown in figure 20, which also gives the W/L ratios for the gates and switches.

Figure 14 shows the model plotted as a function of V_{ds}, with several relevant values of the parameter $(V_{gs}-V_{th})$. This model was used directly for the enhancement type of series switching transistors.

An enhancement mode and a depletion mode model were combined to produce an inverting logic gate, where the output nett current is a function of the supply voltage (assumed to be 5.00), the output voltage, the enhancement mode transistor gate voltage, and the value of the gate size ratios, W/L, for the two transistors. Figure 15 shows the model characteristics for a gate size ratio of 4, and a square channel area for the pull-down transistor. Figure 16 shows the same characteristics, but for a gate size ratio of 8. It is clear that, over the main range of gate and output voltages, the output impedance of the gate is quite high: much higher than the 10 kΩ often assumed for rule of thumb time delay estimations. Thus these estimates are likely to be somewhat optimistic.

18.3 Clock Splitter delays

(a) Tie-busses as delay lines

It was assumed (Sub-section 15.3) that the resistance of the vertical diffusion layer tie-busses, in conjunction with the lumped capacity of the Array clock rails (figure 13), would act as delay lines, to give a large part of the delay needed to generate the appropriate operation of the Clock Splitter (Sub-sections 11.2 and 15.1).

Figure 17(a) shows an equivalent circuit with resistance and capacitance values calculated from actual chip areas, and from the n-mos parameters listed in Table 1.

A theoretical analysis of the transient response of this circuit, assuming a step function of voltage at the input and a value of R of 2500 Ω (the assumed output impedance of an inverting super-buffer) showed that there was a predominant time constant of 7.2 ns, with subordinate ones all less than 0.3 ns. The latter have negligible effect at the time of occurence of the half output voltage level.

If all the capacitance is lumped at the centre clock rail, the single time constant is 7.6 ns.

It was concluded, then, that the distributed nature of the tie-busses is negligible (particularly as the output impedance of the super-buffer is likely to be even larger than assumed), and that an adequate approximation is obtained by lumping the total capacitance at the output of the super-buffer.

(b) Inverting super-buffer delays

Figure 17(b) shows the equivalent circuit derived for an inverting super-buffer. The numbers adjacent to the transistors indicate the W/L ratio. The capacitances are derived from the actual chip areas involved. This circuit was simulated in a computer program, using the gate model discussed above, and with a time increment of 0.1 ns. The time delay to the half voltage output point, for a step function input, was determined for both directions of step, and for output capacitances of 0.29 pF (representing A_1 of figure 12(b)), and of 2.62 pF (representing ISB_{2_b} of figure 12(b)).

Results are shown in figure 18, where the numbers above and below the delay elements give the delay (ns) for rising and falling logic levels, respectively.

operating speed being left rather to chance. It is nevertheless important that the speed of operation of the various circuits be estimated, for comparison with the measured performance, as a guide to the adequacy of the estimation assumptions.

It is also necessary to show that the magnitude of the time delays, essential to the operation of the Clock Splitter (Section 15), are adequate; and that the transfer of data bits between the four main storage capacitances (provided by stray circuit capacity) within each cell of the Two Dimensional Shift Register Array, takes place in a reliable and predictable manner.

Although the general circuit simulation program, SPICE, and the simulation programs specifically slanted towards n-mos VLSI circuits, ESIM and TSIM, were available on the VLSI Program computer, there was simply no time to put them to use before the closing date for the chip design data. Thus there was no choice but to rely on instinct and judgement, derived from previous experience with similar designs in discrete component circuits, when producing the final design. (It was, no doubt, for this reason that the VLSI Program strongly discouraged students from attempting to use sophisticated circuits: such as the library cell clock-splitter).

The estimates described in this Section were all made after the chip design was frozen (and, in fact, after the chip had been tested and been found to operate as planned). Because the time element was then no longer critical, opportunity was taken to examine the operation of the various circuits in some detail, so as to gain a greater insight into the advantages and limitations of the n-mos technology, rather than to use a high power simulation program which merely gives a go or no-go answer to a particular design. To this end, some simple circuit simulation programs were written (in the PASCAL language), and some of the RC delay networks were examined by theoretical analysis. In other cases, rules of thumb for time delays, given by Mead and Conway, were used.

18.2 Model of n-mos FET transistor

For lack of better information, the model, chosen for the current-voltage relationships in both the enhancement and depletion modes of n-mos transistor, is a hyperbolic tangential transition between assymptotic relationships defined by Mead and Conway (equations 1 - 6 and 1 - 3) for the fully saturated and fully unsaturated conditions of operation. This model is defined by:

$$\left.\begin{aligned} I_{ds} &= \frac{\mu\epsilon}{2D}\cdot\frac{W}{L}\cdot(V_{gs}-V_{th})^2\cdot\tanh(2.V_{ds}/(V_{gs}-V_{th})) \;;\; V_{gs} > V_{th} \\ I_{ds} &= 0 \;;\quad V_{gs} \leq V_{th} \end{aligned}\right\} \qquad (6)$$

where

(a) the voltage suffixes, g, d and s refer to gate, drain and source

(b) V_{th} is the threshold voltage (0.8 or -3.0 V, from Table 1)

(c) $\mu.\epsilon/2D$ is the conduction factor (taken as 12 $\mu A/V^2$ - the mean of the measured AMI and Comdial values - Table 1)

(d) W, L are the width and length of the transistor gate

For the Clock Splitter, the current depends on the level of the clock input signal. Table 6 lists the contributions to the value. The element designations of figure 12(b) apply.

Table 7 lists the contributions to the current of the Clock Selector Switch. All four clock inputs are assumed to be at the same logic level. The control signal, Enable X-Shift, has a small effect on the current.

Table 8 lists the contributions to the current of the Data Output Selector. Two input signals and three control signals affect the consumption. A listing of all possible combinations gives the maximum, minimum and average (assuming equal weight is allocated to each possible input combination) values, shown in the Table.

Summing the maximum, minimum and average values (including the special average value for the Data Output Selector) for two Arrays, two Clock Splitters and one each Clock Selector Switch and Data Output Selector, the total width/length value for the MPC chip is (depending on the state of the clocks, data and control signals):

maximum W/L = 168.5
average W/L = 136.8
minimum W/L = 102.1

Combining these values with the values, 150, 125 and 100 μA per square (from the previous section), the current estimates become:

maximum = 25.3 mA
average = 17.1 mA
minimum = 10.2 mA

17.3 Charging current

A rough estimate for the area and length of the four significant layers (metal, diffusion, polysilicon and channel), driven (internally) by clock signals, has been made (by scaling from the PLOTCIF diagrams), and the results are given in Table 9. The effective capacity is also listed, based on the capacitances given in Table 1. The total capacitance is of the order of 23 pF. Assuming a maximum clock frequency of 1 MHz on both Arrays simultaneously, the charging current on the clock capacitance is of the order of 115 μA. This is clearly negligible compared with the saturation current.

The charging current for signals passing through the Arrays is also negligible, particularly since it is very small unless a mix of Highs and Lows are passing. In this case, the saturation current is reduced very significantly, more than cancelling the additional charging current.

Thus the charging current is entirely negligible, as far as capacities on the chip are concerned. Capacitance connected externally via the output pads is also likely to be well under 1 mA (200 pF at 1 MHz).

18. ESTIMATION OF OPERATING SPEED

18.1 General

The envisaged application of the Serial Data Matrix Transposer requires only a modest clock rate; say 150 kHz, when reading or writing the multiplexed data. This is well within the capability of the n-mos process. Thus economy of current consumption was chosen as the main design requirement, with

16.3.4 PROJECT is the complete chip design, calling up a normal and a mirror image SRARRAY; adding two normal (see Sub-section 15.4) instances of SB2IF; with PADGROUND, CLOCKSB, CLOCKSEL, PLAPROJ and pads along the right-hand-side; and calling up all interconnections to these elements.

16.4 Chip area

The project is contained within a rectangular area, 1160x920λ, or 2920x2300 μm, and has 22 pads to be connected to package pins (in addition to the substrate connection).

17. ESTIMATION OF CHIP CURRENT CONSUMPTION

17.1 General method

Digital logic circuits, realised in n-mos technology, consume current from the power supply in two ways. Firstly, there is a continuous current flowing through the saturated depletion-mode current-source transistors associated with each gate with a turned-on pull-down switch-transistor. Secondly, there is a charging current required to charge the circuit capacitances under dynamic operating conditions, and dependent on the rate of the clock driving the circuit.

An estimate of the saturation current is given by equation (2) and Sub-section 5.2(a). An appropriate range of values would be 100 to 150 μA per square of depletion transistor channel area. The total saturation current for the MPC chip is therefore estimated by accumulating the width/length values for all current sources which are associated with turned-on pull-down transistors, and multiplying the result by a factor from the above range. Several conditions of clock state and control signal state may need separate consideration.

The charging current is estimated by:

$$I_c = f.C.\Delta V \qquad (5)$$

where C is the capacity of the channel gate and wiring
ΔV is the change in logic voltage
f is the frequency of the relevant clock

For values of 1 pF, 5 V and 1 MHz, the current is 5 μA.

17.2 Saturation current

Table 4 lists the accumulated width/length values for the basic elements used in the MPC design. These are taken from scaled plots (obtained via the program PLOTCIF) of the elements. In most cases, different values apply depending on the logical level assumed at the output of the element (ie on whether the pull-down transistors are turned on or off). Some elements are symmetrical in the sense that one of two identical channels is always conducting.

For the Shift Register Array, the maximum current is drawn when the cells all hold the Low logic level (determined by the raster-return non-inverting super-buffers). Due to logical inversion at the inputs and output of the Array, this corresponds to an Array output at the logical High level. Table 5 lists the contributions to the Array value.

I.3 Execution

(1) Edit READTAPE.CNTL at items (1), (6), (7), (8) and (15) as required, ensuring that the dataset (15) does not already exist.

(2) Supply tape (6) to the Computing Office.

(3) In TSO mode, enter "Submit Readtape".

(4) Enter "Status". A reply "On Output Queue" indicates that the job is complete.

(5) Retrieve the tape from the Computing Office.

APPENDIX II

REFORMATTING THE BLOCK-DATA FOR DISPLAY

II.1 Source program "FILPLOT"

```
PROGRAM FILPLOT(INFILE,OUTFILE,OUTPUT);
VAR
  INFILE,OUTFILE,OUTPUT  : TEXT;
  CH                     : CHAR;
  I,J,L,M,N              : INTEGER;
  VALID                  : BOOLEAN;
BEGIN
  TERMOUT(OUTPUT);
  RESET(INFILE);
  REWRITE(OUTFILE);
  VALID := TRUE;
  WHILE VALID AND NOT EOF(INFILE) DO
    BEGIN
    FOR I:= 1 TO 72 DO
    IF NOT EOF(INFILE) THEN
      BEGIN
      READ(INFILE,CH);
      WRITE(OUTFILE,CH);
      END;
    WRITELN(OUTFILE);
    END;
  CLOSE(INFILE);
  CLOSE(OUTFILE);
  WRITELN(OUTPUT,'PLOT FINISHED');
END.
```

II.2 CLIST(FILPLOT)

```
PROC 2 D Q
ATTR MYATTR RECFM(F B) DSORG(PS) LRECL(80) BLKSIZE(3600)
ALLOC DA(&D) FI(INFILE)
ALLOC DA(&Q) FI(OUTFILE) NEW USING(MYATTR)
PASCALVS FILPLOT
PASCMOD FILPLOT
CALL FILPLOT
FREE FI(INFILE)
FREE FI(OUTFILE)
FREE ATTRLIST(MYATTR)
DEL FILPLOT.LOAD
DEL FILPLOT.OBJ
```

II.3 Notes on parameters

(1) INFILE is the DDname for the file containing the input data in blocked format (eg PLAGEN.BLK). It replaces the parameter D.

(2) OUTFILE is the DDname for the output file (eg PLAGEN.PLT) to be displayed as required. It replaces the parameter Q. It must not exist prior to the execution of the CLIST as it is specified to be NEW.

(3) OUTPUT is the DDname for the message file (ie the terminal screen).

(4) VALID is ineffective.

APPENDIX III

REFORMATTING THE BLOCK DATA TO LINE DATA

III.1 Source program "FILFORM"

```
PROGRAM FILFORM(INFILE,OUTFILE,OUTPUT);

LABEL 100;

CONST  UCASE = TRUE;

VAR
  INFILE,OUTFILE,OUTPUT  : TEXT;
  CH                     : CHAR;
  I,J,K,L,M,N              : INTEGER;
  VALID,FLAG             : BOOLEAN;
  Z                      : ARRAY(.1..220.) OF CHAR;

BEGIN
  TERMOUT(OUTPUT);
  RESET(INFILE);
  REWRITE(OUTFILE);
  VALID := TRUE;
  IF EOF(INFILE) THEN WRITELN(OUTPUT,'INFILE EMPTY');
  WHILE VALID AND NOT EOF(INFILE) DO
    BEGIN
    WHILE NOT EOF(INFILE) AND ((INFILE@ =' ') OR (INFILE@ ='¬'))
         DO GET(INFILE);
    IF EOF(INFILE) THEN GOTO 100;
    I := 1; L := 0;
    WHILE (I<5)AND NOT EOF(INFILE) DO
      BEGIN
      N := ORD(INFILE@);
      GET(INFILE);
      IF EOF(INFILE) AND (I<4) THEN VALID := FALSE;
      IF (N<240) OR (N>249) THEN VALID := FALSE
        ELSE L := L*10+N-240;
      I := I+1;
      END;
    L := L-4;
    IF (L<0) THEN VALID := FALSE;
    IF (L=0) THEN WRITELN(OUTFILE);
    J := 0;
    WHILE VALID AND (L>0) AND (J<219) AND NOT EOF(INFILE) DO
      BEGIN
      J := J+1;
      N := ORD(INFILE@);
      IF UCASE AND ((128<N) AND (N<138) OR (144<N) AND (N<154)
               OR (161<N) AND (N<170)) THEN N := N+64;
      IF (N=95) THEN N:=124;
      IF (N=192) OR (N=74) THEN
        BEGIN
        Z(.J.) :='(';
        IF (N=192) THEN N:= ORD('*') ELSE N:= ORD('.');
        J := J+1;
        END;
      IF (N=208) OR (N=90) THEN
        BEGIN
        IF (N= 08) THEN Z(.J.):='*' ELSE Z(.J.):='.' ;
        N := ORD(')');
```

```
            J := J+1;
            END;
          Z(.J.) := CHR(N);
          L := L-1;
          GET(INFILE);
          IF EOF(INFILE) AND (L>0) THEN VALID := FALSE;
          END;
        IF (J>218) THEN WRITELN(OUTPUT,'LINE TOO LONG');
        FLAG := FALSE;
        M := 0;
        WHILE (M<J) DO
          BEGIN
          IF NOT FLAG AND ((J-M)<73) OR FLAG AND ((J-M)<56)
                      THEN K := J ELSE
            BEGIN
            IF FLAG THEN K := M+55 ELSE K := M+72;
            WHILE (Z(.K.)<>' ') AND (Z(.K.)<>',') AND (Z(.K.)<>';')
                  AND NOT((Z(.K.)='-') AND (Z(.K-1.)='-'))
                  AND NOT((Z(.K.)='*') AND (Z(.K-1.)='*'))
                  DO K := K-1;
            END;
          IF FLAG AND (M<K) THEN FOR I :=1 TO 17 DO WRITE(OUTFILE,' ');
          IF NOT (M<K) THEN
            BEGIN
            WRITE(OUTPUT,'STRING TOO LONG');
            M := J;
            END;
          WHILE (M<K) DO
            BEGIN
            M := M+1;
            WRITE(OUTFILE,Z(.M.));
            END;
          WRITELN(OUTFILE);
          FLAG := TRUE;
          END;
    100:
        END;
      CLOSE(INFILE);
      CLOSE(OUTFILE);
      IF NOT VALID THEN WRITELN(OUTPUT,'INVALID DATA FORMAT ON FILE')
                ELSE WRITELN(OUTPUT,'FORMAT CHANGE COMPLETED');
    END.
```

III.2 CLIST(FILFORM)

```
PROC 2 D Q
ATTR MYATTR RECFM(F B) DSORG(PS) LRECL(80) BLKSIZE(3600)
ALLOC DA(&D) FI(INFILE)
ALLOC DA(&Q) FI(OUTFILE) NEW USING(MYATTR)
PASCALVS FILFORM
PASCMOD FILFORM
CALL FILFORM
FREE FI(INFILE)
FREE FI(OUTFILE)
FREE ATTRLIST(MYATTR)
DEL FILFORM.LOAD
DEL FILFORM.OBJ
```

III.3 Notes on parameters

(1) INFILE is the DDname for the file containing the input data in blocked format (eg PLAGEN.BLK). It replaces the parameter D.

(2) OUTFILE is the DDname for the output file to contain the lined text-type data (eg PLAGEN.PASCAL). It replaces parameter Q. It must not exist prior to the execution of the CLIST as it is specified to be NEW.

(3) OUTPUT is the DDname for the message file (ie the terminal screen).

(4) Lower to upper case conversion can be inhibited by setting UCASE to FALSE in FILFORM.

III.4 Note on PASCAL strings

The procedure for splitting long lines of text into two or more shorter lines may split a PASCAL string into two parts on different lines. This is not permitted in PASCAL. The split string can be validated by adding a single quotation mark at the end of one line, and another at the beginning of the next. This must be done by manual editing of the output file. The compiler will identify the relevant line numbers.

APPENDIX IV

MODIFICATIONS TO PASCAL SOURCE PROGRAMS

IV.1 BELLE

(1) No split strings required attention.

(2) Procedure Importsymbols (line 1688)

The non-standard VAX method of allocating a dataset to the DDname "CIFIN" from within the user's BELLE Procedure has been deleted. Allocation must be made within the CLIST(BELLE).

(3) Procedure Usercode (line 2024)

The non-standard VAX method of including the user's BELLE Procedure (located in another file) has been changed to the non-standard IBM method "%INCLUDE".

(4) Main Program (line 2102)

The allocation of the DDname "OUTPUT" to the VDU Terminal has been carried out from within the program using the non-standard IBM method "TERMOUT".

(5) Procedure Putmsg (line 243)

Minor changes were made to the second "While" statement.

(6) Procedure Makecif (line 499)

The IBM system does not support the TAB control character on its output devices. A modification has been incorporated to simulate the function by identifying TAB with the printing of 8 spaces.

IV.2 PLOTCIF

(1) Ten cases of split strings were corrected.

(2) Three other minor problems were corrected.

The program now compiles correctly, but has not been used.

IV.3 PLAGEN

(1) Ten cases of split strings were corrected.

(2) Main Program (line 767)

The allocation of the VDU terminal to the DDnames "INPUT" and "OUTPUT" is carried out from within the program using the non-standard IBM procedures "TERMIN" and "TERMOUT".

(3) Main Program (line 767)

The arrangement for printing the date has been modified.

(4) Procedure Inputparameters (line 180)

Ten instances of the statement "WRITE" were converted to the statement "WRITELN".

(5) Function Readynln (line 164)

One instance of the statement "WRITE" was changed to the statement "WRITELN". One set of capitals "Y, N, O" was replaced with the lower case equivalents, to ensure that the program accepts lower case inputs at the VDU terminal.

(6) Procedure Preprocess (line 712)

The declaration of the loop-control variables was added.

IV.4 GETSYMBOLS

(1) Five cases of interrupted strings were corrected.

(2) Main Program (line 1451)

The allocation of the VDU terminal to the DDnames "INPUT" and "OUTPUT" is carried out from within the program using the non-standard IBM procedures "TERMIN" and "TERMOUT".

(3) Procedure Printbb (line 75)

A modification to replace the TAB control by the printing of several spaces was added.

APPENDIX V

TRANSFER OF DATA FROM IBM TO VAX VIA MAGNETIC TAPE

V.1 JCL sequence "WRITAPE.CNTL"

```
//JWHZ JOB ,,CLASS=1,MSGCLASS=Z,NOTIFY=JWH
//        EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=Z
//SYSUT1   DD DSN=JWH.PROJ.DATA,DISP=OLD
//SYSUT2   DD UNIT=TAPE1600,DISP=OLD,
//            VOL=(PRIVATE,RETAIN,,,SER=HAYWA1),
//            DCB=(DEN=3,BLKSIZE=1600,RECFM=FB,OPTCD=Q,LRECL=80),
//            LABEL=(1,NL)
//SYSIN    DD DUMMY
//        EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=Z
//SYSUT1   DD DSN=JWH.PROJ.DATA,DISP=OLD
//SYSUT2   DD UNIT=TAPE1600,DISP=OLD,
//            VOL=(PRIVATE,,,,SER=HAYWA1),
//            DCB=(DEN=3,BLKSIZE=1600,RECFM=FB,OPTCD=Q,LRECL=80),
//            LABEL=(2,NL)
//SYSIN    DD DUMMY
```

V.2 Notes on parameters

(1) JWH is the user's user-identification.

(2) IEBGENER is the IBM utility program which writes the tape.

(3) SYSPRINT is the DDname of the message dataset required by IEBGENER.

(4) SYSUT1 is the DDname of the input dataset required by IEBGENER.

(5) JWH.PROJ.DATA is the name of the file (in the IBM system) to be written on the tape.

(6) SYSUT2 is the DDname of the magnetic tape, to be written upon by the program IEBGENER.

(7) TAPE 1600 defines the device (tape drive operating at 1600 bits/inch).

(8) HAYWA1 is the name written on the reel of magnetic tape, identifying it to the system operator.

(9) DEN=3 is a sub-parameter specifying 1600 bits/inch, 9-tracks.

(10) OPTCD=Q specifies that translation from EBCDIC to ASCII is required.

(11) 1, 2 represent the sequence numbers of the files appearing on the tape.

(12) NL specifies that the tape has no labels.

(13) RETAIN prevents the tape being rewound and removed from the tape device before the second version of the file is written.

V.3 Execution

(1) Obtain a suitable blank tape with a write-permit ring in place.

(2) Label it with a name, both externally and magnetically (the latter is both required and carried out by the IBM system Operator).

(3) Edit WRITAPE.CNTL at items (1), (5) and (8), as required.

(4) Ensure that the file to be written is un-numbered and in a suitable format.

(5) Supply the tape (8) to the Computing Office.

(6) In TSO mode, enter "SUBMIT WRITAPE".

(7) Enter "STATUS", to obtain an indication of the progress of the job.

(8) Retrieve the tape (8) from the Computing Office.

APPENDIX VI

USING THE VLSI UTILITY PROGRAMS PLAGEN, GETSYMBOLS AND BELLE

VI.1 Source programs

The PASCAL source programs PLAGEN, GETSYMBOLS and BELLE are stored as members (with corresponding names) of the partitioned dataset VLSI.UTLIB.

VI.2 CLIST

The CLIST members, PLAGEN, GETSYM and BELLE, listed below, are required to execute the programs.

CLIST(PLAGEN)

```
PROC 1 PLAMEMB
COPY VLSI.UTLIB(PLAGEN) PLAGEN.PASCAL
PASCALVS PLAGEN
PASCMOD PLAGEN
ATTR MYATTR RECFM(F B) DSORG(PS) LRECL(80) BLKSIZE(3600)
ALLOC DA(VLSI.PLALIB(&PLAMEMB)) FI(PLAFILE) OLD
ALLOC DA(VLSI.CIFLIB(&PLAMEMB)) FI(CIFFILE) OLD
ALLOC DA(PLAGEN.TEM1) FI(ORPLANE) NEW USING(MYATTR)
ALLOC DA(PLAGEN.TEM2) FI(GENFILE) NEW USING(MYATTR)
CALL PLAGEN
FREE FI(PLAFILE)
FREE FI(CIFFILE)
FREE FI(ORPLANE)
FREE FI(GENFILE)
FREE ATTRLIST(MYATTR)
DEL PLAGEN.*
```

CLIST(GETSYM)

```
PROC 1 CIFMEMB
COPY VLSI.UTLIB(GETSYM) GETSYM.PASCAL
PASCALVS GETSYM
PASCMOD GETSYM
ALLOC DA(VLSI.CIFLIB(&CIFMEMB)) FI(CIFFILE) OLD
ALLOC DA(VLSI.HDRLIB(&CIFMEMB)) FI(HDRFILE)
CALL GETSYM
FREE FI(CIFFILE)
FREE FI(HDRFILE)
DEL GETSYM.*
```

CLIST(BELLE)

```
PROC 2 BELMEMB CIFMEMB
COPY VLSI.UTLIB(BELLE) BELLE.PASCAL
COPY VLSI.PROJLIB(&BELMEMB) BELTEM.PASCAL(BELPROC)
PASCALVS BELLE LIB(BELTEM.PASCAL)
PASCMOD BELLE
ALLOC DA(VLSI.HDRLIB(&CIFMEMB)) FI(CIFIN) OLD
ALLOC DA(VLSI.CIFLIB(&BELMEMB)) FI(CIFFILE)
CALL BELLE
FREE FI(CIFFILE)
FREE FI(CIFIN)
DEL (BELLE.*,BELTEM.PASCAL(BELPROC),BELTEM.PASCAL)
```

VI.3 Design datasets

The user's designs (in the form of BELLE-code to be called up by the program BELLE) are stored as members of the partitioned dataset VLSI.PROJLIB.

The user's designs of programmable logic arrays (in the form specified as input to the program PLAGEN) are stored as members of the partitioned dataset VLSI.PLALIB.

VI.4 CIF library of symbols

The partitioned dataset VLSI.CIFLIB contains as members, the CIF-code defining:

(a) the standard cells provided by the VLSI Program.

(b) programmable logic array cells, produced previously by the user with the program PLAGEN.

(c) user designed cells produced previously by the user with the program BELLE.

VI.5 Header list of CIF symbols

The partitioned dataset VLSI.HDRLIB contains the output of the GETSYMBOLS program, being key information on symbols held in CIF-code form, and called up by the BELLE program.

VI.6 Producing a chip design in CIF-code

(a) Generate the specifications for the required programmable logic arrays and store them as separate members of VLSI.PLALIB.

(b) Generate the CIF-code for each PLA in turn, using the CLIST member PLAGEN (the parameter being the appropriate member name of VLSI.PLALIB). The CIF-code is stored in VLSI.CIFLIB with the corresponding member name.

(c) Merge into one member of VLSI.CIFLIB, copies of the CIF-code for all the cells, standard and user generated, that are to be called up in the chip design.

(d) Generate the header list for this sub-library using the CLIST member GETSYM (the parameter being the member name of the merged dataset produced in (c) above). The header list appears as the corresponding member of VLSI.HDRLIB.

(e) Generate the main chip design in BELLE-code, calling at will on those symbols contained in (c) above.

(f) Generate the new CIF-code for the design, using the CLIST member BELLE. The parameters are:

(1) the member name of VLSI.PROJLIB containing the BELLE-code

(2) the member name of the merged CIF-code file generated in (c) above.

The new CIF-code appears in VLSI.CIFLIB with the same member name as that of parameter (1).

(g) Produce the full CIF-code design file, by merging the old code from (c) above, ahead of the new code from (f) above. Any spurious CIF statements such as duplicated λ definitions, or END statements should be edited out of the file, to produce a legal CIF design.

(h) The design is now ready for checking using the program PLOTCIF to plot all or parts, or using other design checking programs held by the VLSI Program.

(i) The checked design is transmitted to the VLSI Program via magnetic tape, as described in Section 8.

APPENDIX VII

PROJECT-DEFINING BELLE-CODE

```
VAR
    I,J,X,Y,Z,X1,X2,Y1,Y2 : INTEGER ;
BEGIN
  IMPORTSYMBOLS;
  SETSYMNO(1500);
  DEFINE('PULL4');
    DP(0,0,90);
   LAYER(POLY);
    BOX(-3,0,3,9);
   LAYER(IMPLANT);
    BOX(-3,-2,3,11);
  ENDDEF;
  DEFINE('SBMOD1'); (* ADD GATES TO SB1.RAILS NOT EXTENDED.*)
    DRAW('INVERTINGSBPAIR',0,0);
    DRAW('PULL4',-9,49);ROT(180);
    DRAW('PULL4',-1,43);ROT(180);
    DRAW('PULL4',37,43);ROT(180);
    MD(-15,13);
    MD(-15,15);
    MD(-9,30);
    MD(-1,30);
    MD(37,30);
    MP(14,47);
    MP(22,47);
    MP(6,55);
    MP(30,55);
    MP(46,55);
   LAYER(DIFFUSION);
    BOX(-16,16,-14,53);
    BOX(-2,61,38,63);
    BOX(-10,31,-8,53);
    BOX(-16,51,-7,53);
    BOX(-2,31,0,63);
    BOX(36,31,38,63);
    BOX(9,53,27,57);
    BOX(16,56,20,63);
    BOX(-16,23,1,25);
    BOX(-1,11,1,25);
   LAYER(POLY);
    BOX(14,44,16,46);
    BOX(20,44,22,46);
    BOX(-7,47,2,49);
    BOX(4,56,8,60);
    BOX(4,58,15,60);
    BOX(13,51,15,60);
    BOX(21,51,23,60);
    BOX(21,58,32,60);
    BOX(28,56,32,60);
    BOX(-18,19,-9,21);
    BOX(-11,0,-9,21); (* X *)
    BOX(-6,0,-4,27); (* Y *)
    BOX(34,54,46,56); (* PHI2 *)
   LAYER(METAL);
    BOX(-3,45,5,49);
    BOX(31,45,39,49);
    BOX(15,53,21,56);
    BOX(28,55,32,81);
```

```
NDDEF;
EFINE('SBMOD2'); (* ADD GATES TO SB2.RAILS NOT EXTENDED. *)
 DRAW('INVERTINGSBPAIR',0,0);
 MP(14,47);
 MP(22,47);
 MP(5,54);
 MP(31,54);
LAYER(DIFFUSION);
 BOX(-1,11,2,50);
 BOX(-1,46,11,50);
 BOX(9,53,27,57);
 BOX(25,46,37,50);
 BOX(34,11,37,50);
LAYER(POLY);
 BOX(14,44,16,46);
 BOX(20,44,22,46);
 BOX(5,44,7,53);
 BOX(29,44,31,53);
LAYER(METAL);
 BOX(0,53,21,56);
 BOX(15,46,36,49);
NDDEF;
EFINE('SB2IF');
 DRAW('PULL4',15,62);ROT(90);
 DRAW('PULL4',23,70);ROT(90);
 MD(-6,30);
 MP(29,95);
LAYER(DIFFUSION);
 BOX(-6,31,-4,71);
 BOX(34,49,37,72);
 BOX(-6,61,37,63);
 BOX(-6,69,23,71);
 BOX(25,68,37,72);
LAYER(POLY);
 BOX(13,51,15,60);
 BOX(21,51,23,68);
 BOX(28,66,30,95),
NDDEF;
EFINE('CLOCKSB');
 DRAW('NONINVERTINGSBPAIR',0,0);
 MP(14,54);
 MP(22,54);
 MD(10,-6);
 MD(26,-14);
LAYER(POLY);
 BOX(14,44,16,53);
 BOX(20,44,22,53);
LAYER(DIFFUSION);
 BOX(18,39,26,44);
 BOX(9,-5,11,1);
 BOX(25,-13,27,1);
LAYER(METAL);
 BOX(-9,0,-3,32);
 BOX(-9,0,1,6);
 BOX(-9,28,1,32);
NDDEF;
EFINE('PADINM');
 (* ADD GUARD RING TO PADIN.*)
 DRAW('PADIN',0,0);
 MD(96,111);
 MD(2,2);
```

.E 7. DEPLETION MODE CHANNEL SIZE FOR THE CLOCK SELECTOR SWITCH

Element	Clock input logic state		
	Low	High	
	EXS=X	EXS=Low	EXS=High
Input Inverters (4)	0.000	0.940	0.940
Control Buffers (3)	0.705	0.705	0.705
Output Inverters (2)	0.470	0.000	0.235
Output Pad	8.000	8.000	8.000
Total	9.180	9.650	9.880

.E 8. DEPLETION MODE CHANNEL SIZE FOR THE DATA OUTPUT SELECTOR

Element	Maximum	Minimum	Mean
Input Inverters (5)	5.000	5.000	5.000
Minterm Gates (5)	2.222	1.333	1.887
Output Stages (2)	2.000	0.888	1.583
Output Pad	8.000	8.000	8.000
Tristate Pad	16.000	11.000	14.750
Total	33.220	26.220	31.220

TABLE 9. ESTIMATE OF CAPACITY DRIVEN BY CLOCK WAVES

Layer	Area λ^2	Length λ	Capacity pF
Metal	67 000	-	12.56
Diffusion	6 200	2600	6.48
Polysilicon	1 500	-	0.38
Channel	1 450	-	3.63
Total			23.10

TABLE 10. CLOCK SPLITTER ESTIMATED TIME SEQUENCE

State change	Time (ns)	State change	Time (ns)
clock ↑	0	clock ↓	0
ϕ2b ↓	11.2	ϕ1 ↓	18.2
ϕ2a ↓	11.7	ϕ2a ↑	54.6
ϕ1 ↑	31.9	ϕ2b ↑	78.3
end	45.4	end	78.3

TABLE 4. DEPLETION MODE CHANNEL SIZE FOR BASIC ELEMENTS

Element	Output logic state		Remarks
	Low	High	
Input Pad	0.000	0.000	
Output Pad	8.000	8.000	
Tristate Pad (enabled)	11.000	11.000	
Tristate Pad (not-enabled)	16.000	16.000	
Inverting Super-buffer	1.5000	1.000	
Noninverting Super-buffer	2.000	0.500	
Pull4	0.235	0.000	8.5x2λ channel
Shift-Register Input Gate	0.118	0.000	17x2λ channel
Shift-Register Cell	0.125	0.118	17x2 and 16x2λ channels
PLA Input Inverter	1.000	1.000	2x2λ channels cascaded
PLA Minterm Gate	0.444	0.000	4.5x2λ channel
PLA Output Gate	1.000	0.444	4.5x2 and 2x2λ channels

TABLE 5. DEPLETION MODE CHANNEL SIZE FOR A SHIFT REGISTER ARRAY

Element	Array output logic state	
	Low	High
Input Gates (2)	0.000	0.236
SR Cells (54)	6.372	6.750
Raster NISB (19)	9.500	38.000
Output ISB	1.500	1.000
Output Pad	8.000	8.000
Total	25.370	53.990

TABLE 6. DEPLETION MODE CHANNEL SIZE FOR A CLOCK SPLITTER

Element	Array output logic state	
	Low	High
Input NISB	2.000	0.500
ISB1	3.000	2.500
A1 (ISB)	1.000	1.500
A2	0.236	0.236
OR, A3	0.236	0.236
ISB2A	1.000	1.500
A4	0.236	0.000
ISB2B	1.000	1.500
Total	8.710	7.970

TABLE 2. CONTENTS OF WORKSHOP MAGNETIC COMPUTER TAPE

ɜnce ɔer	File name	Description of contents
1	Preamble	
2	BELLE.DOC	Details of BELLE commands and syntax.
3	BELLEUSE.DOC	Usage of BELLE((General).
4	BELLE.PAS	BELLE Source Code.
5	GETSYMBOL.DOC	Documentation for the Header Extractor.
6	GETSYMBOL.PAS	GETSYMBOLS Source Code.
7	LIBRARY.CIF	CIF-code for the Xerox Cell Library.
8	LIBRARY.DOC	Documentation for the Library Cells.
9	PLAGEN.DOC	Documentation for PLA Generator.
0	PLAGEN.PAS	PLAGEN Source Code.
1	STDLIB.BEL	BELLE-code to generate Lyaer Contacts.
2	STDLIB.CIF	CIF-code from STDLIB.BEL.
3	STDLIB.SLB	Header File from STDLIB.CIF.
4	VIEWCIF.DOC	Documentation for the Plotting Program.
5	VIEWCIF.HLP	More information on Plotting Program.
6	VIEWCIF.PAS	PLOTCIF Source Code.
7	HOME.	Workshop student's Concatenation.
8	BELLE.DOC	Details of BELLE commands and syntax.
9	BELLEUSE.DOC	Usage of BELLE (General).
0	BELLE.PAS	BELLE Source Code.
1	GETSYMBOL.DOC	Documentation for the Header Extractor.
2	GETSYMBOL.PAS	GETSYMBOLS Source Code.
3	LIBRARY,CIF	CIF-code for the Xerox Cell Library.
4	LIBRARY.DOC	Documentation for the Library Cells.
5	PLAGEN.DOC	Documentation for PLA Generator.
6	PLAGEN.PAS	PLAGEN Source Code.
7	STDLIB.BEL	BELLE-code to generate Layer Contacts.
8	STDLIB.CIF	CIF-code from STDLIB.BEL.
9	STDLIB.SLB	Header File from STDLIB.CIF.
0	VIEWCIF.DOC	Documentation for the Plotting Program.
1	VIEWCIF.HLP	More information on Plotting Program.
2	VIEWCIF.PAS	PLOTCIF Source Code.
3	HOME.	Workshop student's Concatenation.

TABLE 3. DATA-FLOW AS A FUNCTION OF CONTROL SIGNAL LEVELS

Enable X-shift	Array Select	Data-In		Raster	Clock	Data-Out		Mode
		Y-To	X-To	A1	A2	X&Y	X	
L	L	A1	-	Y-In	-	-	-	Transmit
H	L	A1	A2*	Y-In	X-Out	A2	A2	
L	H	A2	-	-	Y-In	-	-	
H	H	A2	A1*	X-Out	Y-In	A1	A1	
L	L	A1*	-	Y-Out	-	A1	-	Receive
H	L	A1*	A2	Y-Out	X-In	A1	-	
L	H	A2*	-	-	Y-Out	A2	-	
H	H	A2*	A1	X-In	Y-Out	A2	-	

* Garbage (under normal operation).

TABLE 1. N-MOS PROCESS PARAMETERS

Parameter	Pre AUSMPC 5/82	AMI Chip	COMDIAL Chip	Post AUSMPC 5/82	Units
Threshold:					
enhancement	0.7 to 1.0	0.63	0.95	0.8	V
depletion	-3.5 to -4.5	-2.8	-3.6	-3.0	V
inverter		2.0	2.5		V
Conduction Factor:					
enhancement		13.5	10.5		$\mu A/V^2$
depletion		13.5	10.5		$\mu A/V^2$
Resistivity:					
metal	0.03			0.03	$\Omega/^2$
diffusion	10	8	10	10	$\Omega/^2$
polysilicon	20	20	30	20	$\Omega/^2$
channel	10 000			10 000	$\Omega/^2$
Capacitance:					
metal	0.03			0.03	$fF/\mu m^2$
diffusion	0.1			0.1	$fF/\mu m^2$
polysilicon	0.04			0.04	$fF/\mu m^2$
channel	0.4			0.4	$fF/\mu m^2$
Inverter		0.5	0.6		V
Fully ON		55	75		μA
Ring Osc. Frequency		10	15		MHz
Colour		greenish	blueish		

```
      MP(990,193);
      MP(990,242);
      MP(982,346);
     LAYER(POLY);
      BOX(990,192,1016,194);
      BOX(990,242,988,271);
      BOX(982,345,992,347);
     LAYER(METAL);
      BOX(988,126,992,242);
      BOX(980,126,984,346);
      BOX(998,80,1062,84); (*PLA GND *)
   ENDDEF;
   DRAW('PROJECT',0,0);
END;
```

```
LAYER(DIFFUSION);
 BOX(856,448,860,472); (* PHI-B *)
 MD(858,448);
 MD(858,472);
(* ADD PADS ON RH SIDE. *)
 DRAW('PADINM',1160,602);ROT(90);
 DRAW('PADOUT',1160,140);ROT(90);
 DRAW('PADTRISTATE',1160,352);ROT(90);MX;
 DRAW('PLAPROJ',871,0);
 X:= 928;
 FOR I:= 1 TO 3 DO
   BEGIN
   MP(X,448);
   MP(X,472);
   MP(X,728);
   MP(X,752);
   MP(X,772);
   MP(X,788);
  LAYER(POLY);
   BOX((X-2),448,(X+2),472);
   BOX((X-2),728,(X+2),752);
   BOX((X-2),772,(X+2),788);
  LAYER(METAL);
   BOX((X-2),140,(X+2),448);
   BOX((X-2),472,(X+2),728);
   BOX((X-2),752,(X+2),772);
   BOX((X-2),788,(X+2),808);
   X:=X+16;
   END;
LAYER(METAL);
 BOX(960,696,1049,700); (* PAD IN *)
 FOR I:= 1 TO 2 DO
   BEGIN
     CASE I OF
       1: Y:=321;
       2: Y:=577;
     END;
   MP(960,Y);
   MP(920,Y);
  LAYER(METAL);
   BOX(866,(Y-2),920,(Y+2));
  LAYER(POLY);
   BOX(922,(Y-2),960,(Y+2));
   END;
 Y:= 111;
 Z:= 873;
 FOR I:= 1 TO 4 DO
   BEGIN
     CASE I OF
       1: X:=896;
       2: X:=912;
       3: X:=944;
       4: X:=928;
     END;
  LAYER(METAL);
   BOX(808,(Y-2),Z,(Y+2));
   BOX((Z-2),(Y-2),(Z+2),(Y+10));
   BOX((Z-2),(Y+6),X,(Y+10));
   Y:= Y+7;
   Z:= Z-7;
   END;
```

```
          3: Z:=140;
          4: Z:=133;
          5: Z:=140;
         END;
      LAYER(DIFFUSION);
       BOX((X-3),105,(X+3),114);
      LAYER(POLY);
       BOX((X-1),108,(X+1),Z);
       MP(X,Z);
       X:=X+16;
       END;
     MP(73,140);
    LAYER(METAL);
     BOX(71,133,75,140);
     X:= 111;
     FOR I:= 1 TO 2 DO
       BEGIN
       MP(X,126);
      LAYER(POLY);
       BOX((X-1),120,(X+1),126);
       X:= X+8;
       END;
  ENDDEF;
  DEFINE('PROJECT');
   (* DUPLICATE SR ARRAY,& ADD CONTROL. *)
    DRAW('SRARRAY',170,543);
    DRAW('SRARRAY',170,377);MY;
    DRAW('SB2IF',771,606);ROT(-90);
    DRAW('SB2IF',771,350);ROT(-90);
    DRAW('PADGROUND',0,513);ROT(-90);
   LAYER(METAL);
    BOX(148,454,1060,466); (* VGND CENTRE RAIL. *)
    BOX(79,433,152,487); (* VGND PAD FEED. *)
    BOX(1152,0,1160,920); (* VDD RH EDGE. *)
    BOX(877,814,1062,822);
    BOX(1054,80,1062,908); (* GND RH EDGE *)
    DRAW('CLOCKSEL',1015,920);ROT(180);
    DRAW('PADOUT',1160,814);ROT(90);
    X:= 880;
    Y:= 809;
   LAYER(METAL);
    FOR I:= 1 TO 4 DO
      BEGIN
      BOX(808,(Y-2),(X+2),(Y+2));
      X:= X+10;
      Y:= Y-7;
      END;
   (* ADD CLOCKSB & INPUT PAD. *)
    DRAW('CLOCKSB',996,750);
    DRAW('PADINM',1160,708);ROT(90);
   LAYER(METAL);
    BOX(1016,802,1051,806); (* CORNER TO PHI-B *)
    BOX(799,778,993,782); (* VDD TO CLOCKSB *)
    BOX(950,778,954,808);
    BOX(1031,761,1055,767); (* GND CLOCKSB *)
    BOX(848,742,1006,746); (* PHI-A *)
    BOX(848,726,852,746); (* PHI-A *)
    BOX(856,734,1022,738); (* PHI-B *)
    BOX(856,472,860,738); (* PHI-B *)
    BOX(856,363,860,448); (* PHI-B *)
    BOX(848,363,860,367); (* PHI-B *)
```

```
      LAYER(POLY);
       BOX((X-1),52,(X+1),59);
       BOX((X-1),72,(X+1),79);
       END;
     IF (I=12) THEN
       BEGIN
       MP(X,71);
      LAYER(POLY);
       BOX((X-1),63,(X+1),70);
       END;
     END;
  DP(4,53,180);
  MD(5,100);
  MD(21,66);
  MD(28,46);
  MD(28,76);
  MD(15,116);
  MP(28,83);
  MP(10,66);
  MP(55,112);
  MD(63,112);
  MP(71,112);
  MP(87,112);
 LAYER(METAL);
  BOX(8,64,23,68);
  BOX(26,44,30,85);
  BOX(0,0,138,8); (*VDD *)
  BOX(0,98,138,106); (* GND *)
 LAYER(DIFFUSION);
  BOX(3,41,7,102);
  BOX(14,41,16,115);
  BOX(14,91,35,95);
  BOX(20,55,22,65);
  BOX(53,65,55,107);
  BOX(53,105,64,107);
  BOX(62,105,64,112);
  BOX(74,45,76,57);
  BOX(74,65,76,77);
  BOX(29,45,129,47);
  BOX(20,55,119,57);
  BOX(22,65,109,67);
  BOX(29,75,55,77);
  BOX(74,75,99,77);
  BOX(128,41,132,48);
 LAYER(POLY);
  BOX(-1,52,2,54);
  BOX(1,65,9,67);
  BOX(27,84,29,97);
  BOX(51,87,58,89);
  BOX(49,111,55,113);
  BOX(71,111,76,113);
  BOX(87,111,96,113);
ENDDEF;
DEFINE('PLAPROJ');
  DRAW('PLA500',0,4);
  X:= 25;
  FOR I:= 1 TO 5 DO
    BEGIN
      CASE I OF
       1: Z:=119;
       2: Z:=126;
```

```
BEGIN
X:= I*5;
IF (I=1)OR(I=3)OR(I=7)OR(I=9)OR(I=12)
   OR(I=14)OR(I=16)OR(I=18)OR(I=20)OR
   (I=22)OR(I=24)OR(I=26) THEN
  BEGIN
  DRAW('PULL4',X,39);ROT(180);
  MD(X,6);
 LAYER(DIFFUSION);
  BOX((X-1),7,(X+1),39);
  END;
IF (I=20)OR(I=22)OR(I=24)OR(I=26) THEN
  BEGIN
  Y:= 76-(I-20)*5;
  MD(X,Y);
  MD(X,100);
 LAYER(DIFFUSION);
  BOX((X-2),(Y-2),(X+2),102);
 LAYER(METAL);
  BOX((X-2),36,(X+2),(Y+2));
  Y:= 90-(I-20)*2;
  Z:= 132-((I-20)*7)DIV 2;
  MP((X+5),Z);
 LAYER(POLY);
  BOX((X-4),(Y-1),(X+6),(Y+1));
  BOX((X+4),(Y-1),(X+6),(Z-1));
  END;
IF (I=9)OR(I=14)OR(I=18) THEN
  BEGIN
  DP((X-1),88,180);
  MD(X,100);
 LAYER(METAL);
  BOX((X-2),36,(X+2),90);
 LAYER(DIFFUSION);
  BOX((X-2),86,(X+2),102);
 LAYER(POLY);
  BOX((X-13),87,(X-3),89);
  BOX((X-4),93,(X+6),95);
  BOX((X+4),93,(X+6),113);
  END;
IF (I=7)OR(I=12)OR(I=16) THEN
  BEGIN
  MD(X,83);
  MD(X,100);
 LAYER(METAL);
  BOX((X-2),36,(X+2),85);
 LAYER(DIFFUSION);
  BOX((X-1),81,(X+1),102);
  END;
IF (I=9)OR(I=18) THEN
  BEGIN
  MP(X,51);
  MP(X,71);
 LAYER(POLY);
  BOX((X-1),43,(X+1),50);
  BOX((X-1),63,(X+1),70);
  END;
IF (I=7)OR(I=16) THEN
  BEGIN
  MP(X,51);
  MP(X,71);
```

```
 DRAW('INVERTINGSB',529,-69);
 X:=0;
 FOR I:= 1 TO 18 DO
   BEGIN
   DP((X+5),-6,0);
  LAYER(POLY);
   BOX((X+6),-25,(X+8),-7);
  LAYER(DIFFUSION);
   BOX((X+2),-5,(X+4),1);
   BOX((X+11),-74,(X+13),-68);
   BOX((X+11),-74,(X+28),-72);
   BOX((X+26),-74,(X+28),1);
   X:= X+31;
   END;
 MD(563,9);
LAYER(DIFFUSION);
 BOX(553,-10,564,-8);
 BOX(562,-10,564,8);
LAYER(METAL);
 BOX(558,7,565,11); (* CX *)
 BOX(0,-69,596,-63); (* VDD *)
 BOX(0,-41,596,-37); (* VDD *)
 BOX(-22,-58,558,-52); (* GND *)
(* ADD PADS ON LH SIDE, *)
 DRAW('PADOUT',-170,76);ROT(-90);
 DRAW('PADINM',-170,76);ROT(-90);MX;
 DRAW('PADINM',-170,288);ROT(-90);
 MP(-46,172);
 MP(-46,192);
LAYER(METAL);
 BOX(-170,-84,-162,377); (* VDD *)
 BOX(-72,-84,-64,279); (* GND *)
 BOX(-61,170,-44,174); (* DX *)
 BOX(-61,190,-44,194); (* DY *)
LAYER(POLY);
 BOX(-26,22,-21,24);
 BOX(-45,171,-21,173);
 BOX(-45,191,-21,193);
(* ADD PADS ON TOP. *)
 X:=670;
 Y:=266;
 DRAW('PADVDD',X,377);ROT(180);
 FOR I:= 1 TO 4 DO
   BEGIN
   X:= X-106;
   DRAW('PADINM',X,377);ROT(180);
   MP(590,Y);
   MP(638,Y);
  LAYER(METAL);
   BOX((X-98),(Y-2),(X-94),268);
   BOX((X-98),(Y-2),592,(Y+2));
  LAYER(POLY);
   BOX(591,(Y-1),637,(Y+1));
   Y:= Y-7;
   END;
LAYER(METAL);
 BOX(-170,369,990,377); (* VDD *)
 BOX(-72,271,564,279); (* GND *)
ENDDEF;
DEFINE('CLOCKSEL');
 FOR I:= 1 TO 26 DO
```

```
 MP(588,211);
LAYER(METAL);
 BOX(558,36,590,40);
 BOX(558,57,590,61);
 BOX(558,83,590,87);
 BOX(558,97,590,101);
 BOX(558,159,590,163);
 BOX(558,173,590,177);
 BOX(558,209,590,213);
 BOX(595,-69,607,298); (* VDD *)
(* ADD SBS ON RH SIDE. *)
 DRAW('INVERTINGSB',601,17);ROT(-90);
 DRAW('SBMOD2',601,63);ROT(-90);
 DRAW('NONINVERTINGSB',601,95);ROT(-90);
 DRAW('NONINVERTINGSB',601,165);ROT(-90);
 DRAW('SBMOD1',601,215);ROT(-90);
LAYER(METAL);
 BOX(612,-83,618,232); (*GND *)
 BOX(629,-3,633,298); (* VDD *)
 BOX(653,-13,657,168); (* PHI2 *)
 BOX(646,-6,650,60); (* PHIINV *)
 BOX(646,90,650,98); (* XO *)
 BOX(646,160,650,168); (* XO *)
 BOX(678,10,682,187); (* PHI *)
 BOX(654,183,682,187);
 MP(655,-13);
 MD(648,-7);
 MP(680,12);
 MD(648,99);
 MP(648,89);
 MD(648,169);
 MP(648,159);
LAYER(DIFFUSION);
 BOX(596,6,602,8);
 BOX(596,-8,598,8);
 BOX(596,-8,647,-6);
 BOX(589,36,602,38);
 BOX(587,52,602,54);
 BOX(587,52,589,58);
 BOX(589,84,602,86);
 BOX(589,98,647,100);
 BOX(587,154,602,156);
 BOX(587,154,589,160);
 BOX(587,168,647,170);
 BOX(587,168,589,174);
 BOX(580,188,602,190);
 BOX(576,204,602,206);
LAYER(POLY);
 BOX(645,11,679,13);
 BOX(570,224,602,226 ); (* X *)
 BOX(587,212,589,221); (* Y *)
 BOX(587,219,602,221); (* Y *)
 BOX(574,-14,576,7); (* PHI2 *)
 BOX(574,-14,654,-12); (* PHI2 *)
(* ADD SBS ON BOTTOM. *)
 X:=0;
 FOR I:=1 TO 17 DO
   BEGIN
   DRAW('NONINVERTINGSB',(X+2),-69);
   X:= X+31;
   END;
```

```
 BOX(-9,172,4,176); (* XI *)
 BOX(-22,228,1,232); (* GND *)
LAYER(DIFFUSION);
 BOX(-5,155,-3,180);
 BOX(-11,178,-3,180);
 BOX(-11,178,-9,216);
 BOX(-11,214,-3,216);
 BOX(-5,214,-3,229);
 BOX(-4,215,3,217);
LAYER(IMPLANT);
 BOX(-13,176,-7,218);
 BOX(-13,176,-5,182);
 BOX(-13,212,-5,218);
LAYER(POLY);
 BOX(-22,171,-15,173);
 BOX(-17,168,-15,173);
 BOX(-17,168,-1,170);
 BOX(-13,176,-7,193);
 BOX(-10,172,-7,177);
 BOX(-13,201,-7,218);
 BOX(-10,217,-7,222);
 BOX(-22,191,-15,193);
 BOX(-17,191,-15,226);
 BOX(-17,224, -1,226);
(* ADD LOWEST CARRY-X AT LH. *)
 MP(-10,9);
LAYER(METAL);
 BOX(-11,7,1,11);
LAYER(POLY);
 BOX(-17,8,-11,10);
 BOX(-17,8,-15,24);
 BOX(-22,22,-15,24);
(* ADD RAILS ON RH SIDE. *)
 Y:=0;
 FOR I:=1TO 3 DO
   BEGIN
   MD(563,(Y+59));
   MD(569,(Y+38));
   MD(575,(Y+52));
   MD(581,(Y+31));
  LAYER(METAL);
   BOX(558,(Y+57),565,(Y+61));
   BOX(558,(Y+36),571,(Y+40));
   BOX(558,(Y+50),577,(Y+54));
   BOX(558,(Y+29),583,(Y+33));
   BOX(558,(Y+43),596,(Y+47));
   Y:= Y+76;
   END;
LAYER(DIFFUSION);
 BOX(562,60,564,210);
 BOX(568,39,570,221);
 BOX(574,11,576,203);
 BOX(580,32,582,190);
 MD(588,38);
 MD(588,59);
 MD(588,85);
 MD(588,99);
 MD(588,161);
 MD(588,175);
 DP(569,223,90);
 DP(575,9,-90);
```

```
    BOX(4,39,6,45);
    BOX(18,44,25,46);
    BOX(23,44,25,51);
    BOX(12,49,18,66);
    BOX(15,65,18,70);
    BOX(9,72,24,74);
    BOX(5,60,7,67);
   LAYER(IMPLANT);
    BOX(12,24,18,66);
    BOX(12,60,20,66);
  ENDDEF;
  DEFINE('SRARRAY');
   (* REPLICATE BDSRCELL. *)
    Y:=0;
    FOR I:=1 TO 3 DO
      BEGIN
      X:=0;
      FOR J:=1 TO 18 DO
        BEGIN
        DRAW('BDSRCELL',X,Y);
        X:= X+31;
        END;
      Y:= Y+76;
      END;
   (* ADD TOP CONECTIONS. *)
   LAYER(METAL);
    BOX(0,228,559,232);
    X:=21;
    FOR I:=1 TO 17 DO
      BEGIN
      MD(X,230);
     LAYER(DIFFUSION);
      BOX((X+5),228,(X+14),230);
      X:= X+31;
      END;
    MD(548,230);
   (* ADD LH GND METAL&CARRIES. *)
   LAYER(METAL);
    BOX(-22,-58,-18,279);
    Y:=0;
    FOR I:=1 TO 2 DO
      BEGIN
      MD(-10,(Y+23));
      MD(-10,(Y+85));
     LAYER(DIFFUSION);
      BOX(-11,(Y+24),-9,(Y+84));
     LAYER(METAL);
      BOX(-19,Y,1,(Y+4));
      BOX(-19,(Y+76),1,(Y+80));
      BOX(-12,(Y+21),1,(Y+25));
      BOX(-12,(Y+83),1,(Y+87));
      Y:= Y+76;
      END;
   (* ADD TOP L GATES. *)
    DP(-6,174,180);
    DP(-6,220,180);
    MD(-4,154);
    MD(-10,197);
    MD(-4,230);
   LAYER(METAL);
    BOX(-12,195,1,199); (* VDD *)
```

```
  MD(2,6);
  MD(104,2);
  MD(104,6);
 LAYER(DIFFUSION);
  BOX(95,105,97,110);
  BOX(0,0,4,121);
  BOX(102,0,106,121);
  BOX(0,117,106,121);
ENDDEF;
DEFINE('BDSRCELL');
  DP(20,16,180);
  DP(17,23,0);
  DP(19,68,180);
  DP(11,70,90);
  MD(21,2);
  MD(2,23);
  MD(15,45);
  MD(21,78);
  MP(6,31);
  MP(5,38);
  MP(25,52);
  MP(6,59);
 LAYER(METAL);
  BOX(0,0,32,4);
  BOX(0,7,32,11);
  BOX(0,29,32,33);
  BOX(0,36,32,40);
  BOX(0,43,32,47);
  BOX(0,50,32,54);
  BOX(0,57,32,61);
  BOX(0,76,32,80);
  BOX(14,21,32,25);
 LAYER(DIFFUSION);
  BOX(0,24,2,43);
  BOX(2,0,4,4);
  BOX(2,63,4,77);
  BOX(8,2,10,27);
  BOX(9,25,11,36);
  BOX(9,53,11,68);
  BOX(8,34,10,55);
  BOX(14,8,16,64);
  BOX(20,3,22,10);
  BOX(20,30,22,77);
  BOX(21,14,23,32);
  BOX(26,0,28,49);
  BOX(26,55,28,77);
  BOX(28,47,30,57);
  BOX(2,2,10,4);
  BOX(14,8,22,10);
  BOX(8,18,16,20);
  BOX(0,41,10,43);
  BOX(26,47,30,49);
  BOX(26,55,30,57);
  BOX(14,62,22,64);
  BOX(2,63,11,65);
 LAYER(POLY);
  BOX(12,15,18,17);
  BOX(5,22,12,24);
  BOX(5,22,7,30);
  BOX(17,24,20,28);
  BOX(12,26,18,42);
```

TABLE 11. SHIFT REGISTER CELL TRANSIENT RESPONSE

Cycle	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2
1	0.17	4.99	0.17 3.01	4.99 0.26	4.20 0.28	0.17 3.94	0.17 0.17	4.99 5.00	4.20 4.20	0.17 0.17	0.17 0.17	4.99 5.00	4.20 4.20	0.17 0.17	0.17	4.99
2	0.17	4.99	3.01 4.12	0.26 0.17	0.28 0.17	3.94 4.99	0.17 3.00	5.00 0.26	4.20 0.28	0.17 3.93	0.17 0.17	5.00 5.00	4.20 4.20	0.17 0.17	0.17	4.99
3	0.17	4.99	4.12 4.16	0.17 0.17	0.17 0.17	4.99 5.00	3.00 4.12	0.26 0.17	0.28 0.17	3.93 4.99	0.17 3.00	5.00 0.26	4.20 0.28	0.17 3.93	0.17	4.99
1	4.20	0.17	4.20 0.18	0.17 4.98	0.17 4.03	4.99 0.17	4.20 4.20	0.17 0.17	0.17 0.17	4.99 5.00	4.20 4.20	0.17 0.17	0.17 0.17	4.99 5.00	4.20	0.17
2	4.20	0.17	0.18 0.17	4.98 5.00	4.03 4.12	0.17 0.17	4.20 0.18	0.17 4.98	0.17 4.03	5.00 0.17	4.20 4.20	0.17 0.17	0.17 0.17	5.00 5.00	4.20	0.17
3	4.20	0.17	0.17 0.17	5.00 5.00	4.12 4.15	0.17 0.17	0.18 0.17	4.98 5.00	4.03 4.12	0.17 0.17	4.20 0.18	0.17 4.98	0.17 4.03	5.00 0.17	4.20	0.17

TABLE 12. SHIFT REGISTER CELL TRANSIENT RESPONSE

Cycle	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4*	V1	V2	V3	V4	V1	V2
1	0.28	3.93	0.17 3.00	4.99 0.26	4.20 0.28	0.17 3.93	0.17 0.17	4.99 5.00	4.20 4.20	0.17 0.17	0.17 0.17	4.99 5.00	4.20 4.20	0.17 0.18	0.17	4.99
2	0.17	4.99	3.00 4.12	0.26 0.17	0.28 0.17	3.93 4.99	0.17 2.47	5.00 0.36	4.20 0.40	0.17 3.23	0.17 0.17	5.00 5.00	4.20 4.20	0.18 0.17	0.17	4.99
3	0.17	4.99	4.12 4.16	0.17 0.17	0.17 0.17	4.99 5.00	2.47 4.11	0.36 0.17	0.40 0.17	3.23 4.95	0.17 2.98	5.00 0.26	4.20 0.28	0.17 3.91	0.17	4.99
1	4.03	0.17	4.20 0.18	0.17 4.98	0.17 4.03	4.99 0.17	4.20 4.20	0.17 0.17	0.17 0.17	4.99 5.00	4.20 4.20	0.17 0.17	0.17 0.17	4.99 5.00	4.20	0.17
2	4.12	0.17	0.18 0.17	4.98 5.00	4.03 4.12	0.17 0.17	4.20 0.20	0.17 4.98	0.17 4.03	5.00 0.17	4.20 4.20	0.17 0.17	0.17 0.17	5.00 5.00	4.20	0.17
3	4.15	0.17	0.17 0.17	5.00 5.00	4.12 4.15	0.17 0.17	0.20 0.17	4.98 5.00	4.03 4.12	0.17 0.17	4.20 0.18	0.17 4.98	0.17 4.03	5.00 0.17	4.20	0.17

* C4 = 0.36 pF

TABLE 13. DYNAMIC TEST SPECIFICATION .

Switch positions					FF3 High (TP7*)				FF3 Low (TP7*)			
S1 S2	S3 S4	S5 S6	S7 S8	S9 S10	TP4*	TP5*	TP2	TP1	TP4*	TP5*	TP2	TP1
L H	M X	M X	M L	L H	YX	XY	YX	YX	XY	YX	YX	YX
			H		YX	H	YX	YX	H	YX	YX	YX
				H	H	XY	H	H	XY	H	H	H
X		L	X		H	YY	H	H	YY	H	H	H
			L	X	XX	H	XX	XX	H	XX	XX	XX
H H	M M	X M	X L	L H	YX	XY	XY	$	XY	YX	XY	$
			H		YX	H	H	$	H	YX	H	$
				H	H	XY	XY	$	XY	H	XY	$
X	L		X		H	YY	YY	$	YY	H	YY	$
		L		X	XX	H	H	$	H	XX	H	$

* Actual TP level is inverted.
$ High impedance state.
High impedance if S2 is Low.

During a high impedance state, TP1 is actually Low.

TABLE 14. MEASURED CHIP CURRENTS (mA)

Condition of test	Chip number				
	1	2	3	4	5
5 V: 1 MHz: S1 → L.	17.9	18.9	17.4	30.0	22.9
Minimum operating voltage.	14.7	15.9	14.8	26.3	20.1
4 V	16.8	17.8	16.3	27.9	21.7
6 V	18.7	20.0	18.1	31.3	24.0
S1 → H	18.8	19.9	18.2	30.4	23.8
Test clock 10 kHz.	17.7	18.8	17.2	29.8	22.7
Clockwaves all → Low. S1 → L.	18.9	20.7	19.7	30.8	23.3
Clockwaves all → Low. S1 → H.	19.8	21.8	20.5	31.6	24.2

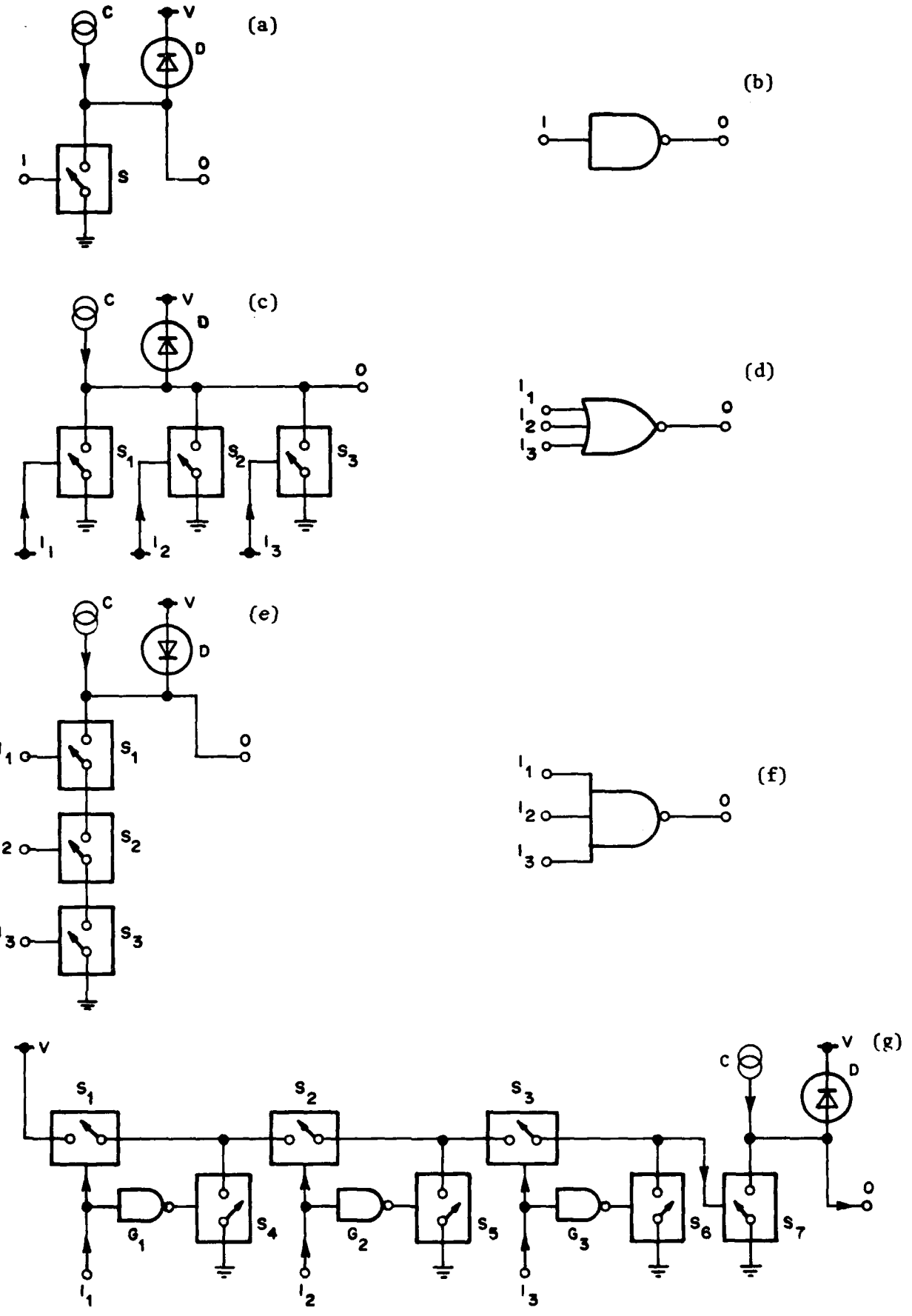


Figure 1. Inverter, NOR and NAND gates

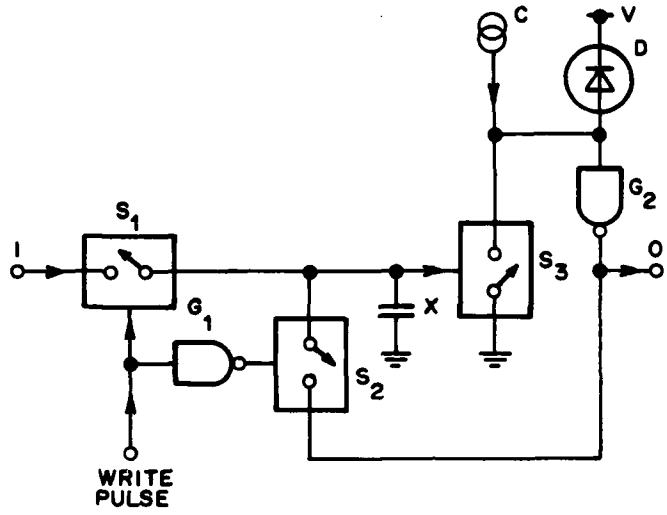


Figure 2. Memory element

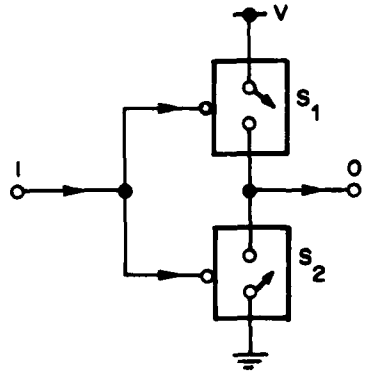


Figure 3. Complementary inverter

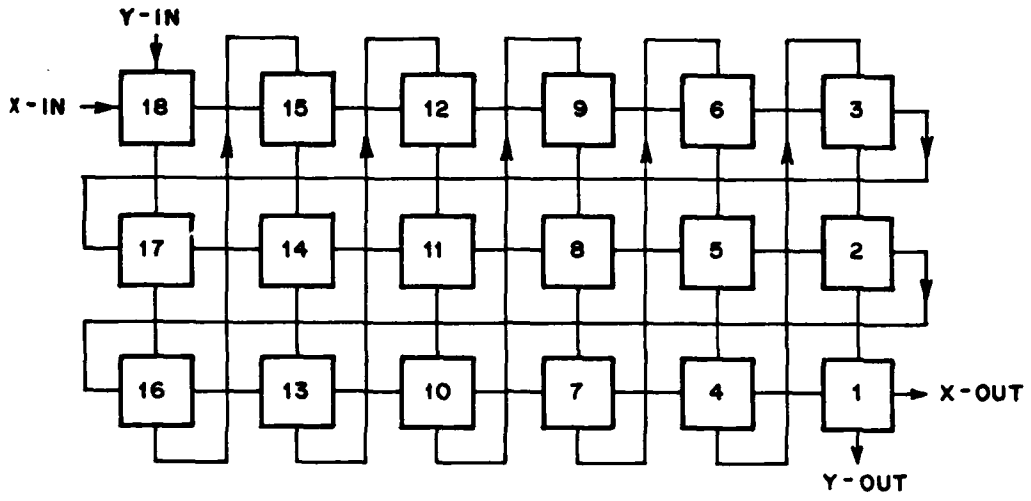


Figure 4. X & Y shift register array

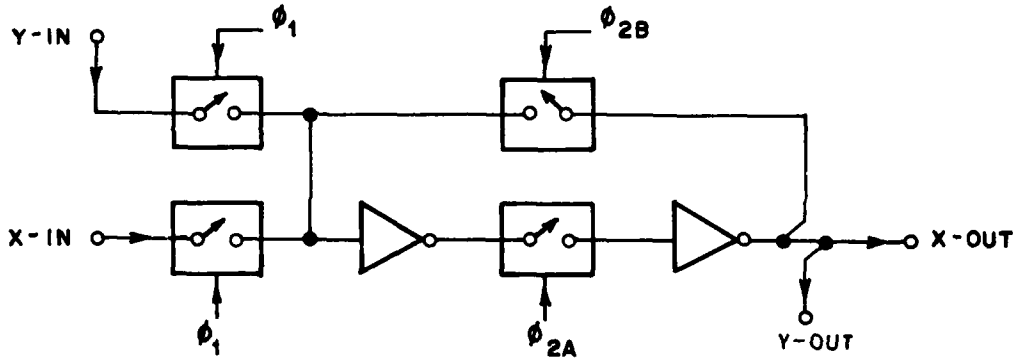


Figure 5. Shift register cell

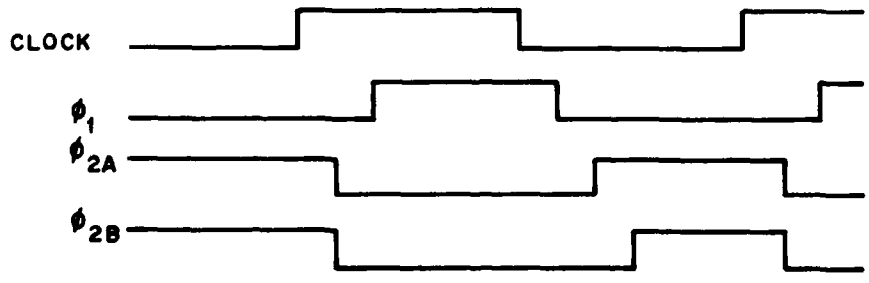


Figure 6. Clock-derived control signals

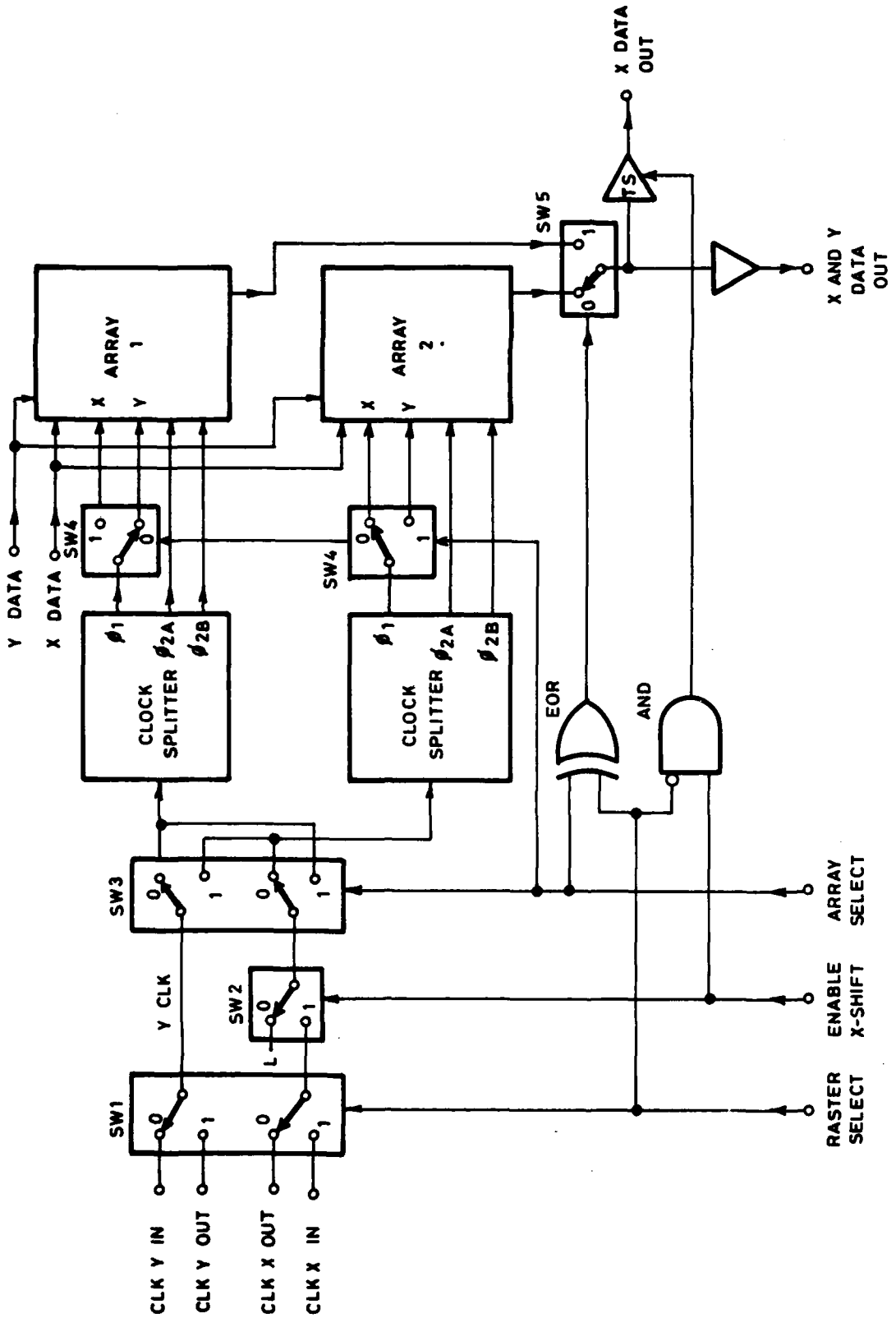
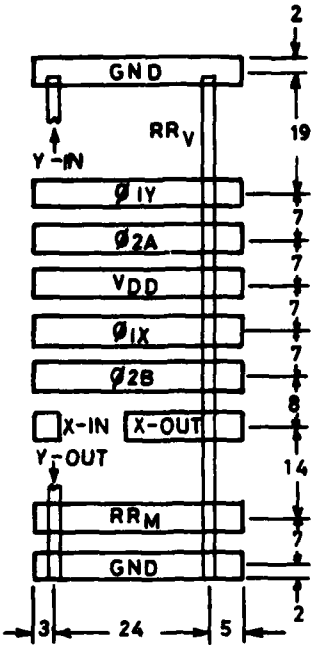
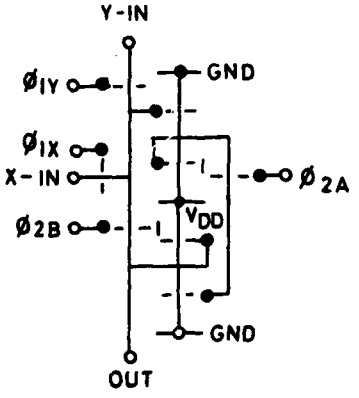


Figure 7. Logic diagram of project



(a) floor plan



(b) stick diagram

Figure 8. Shift register cell

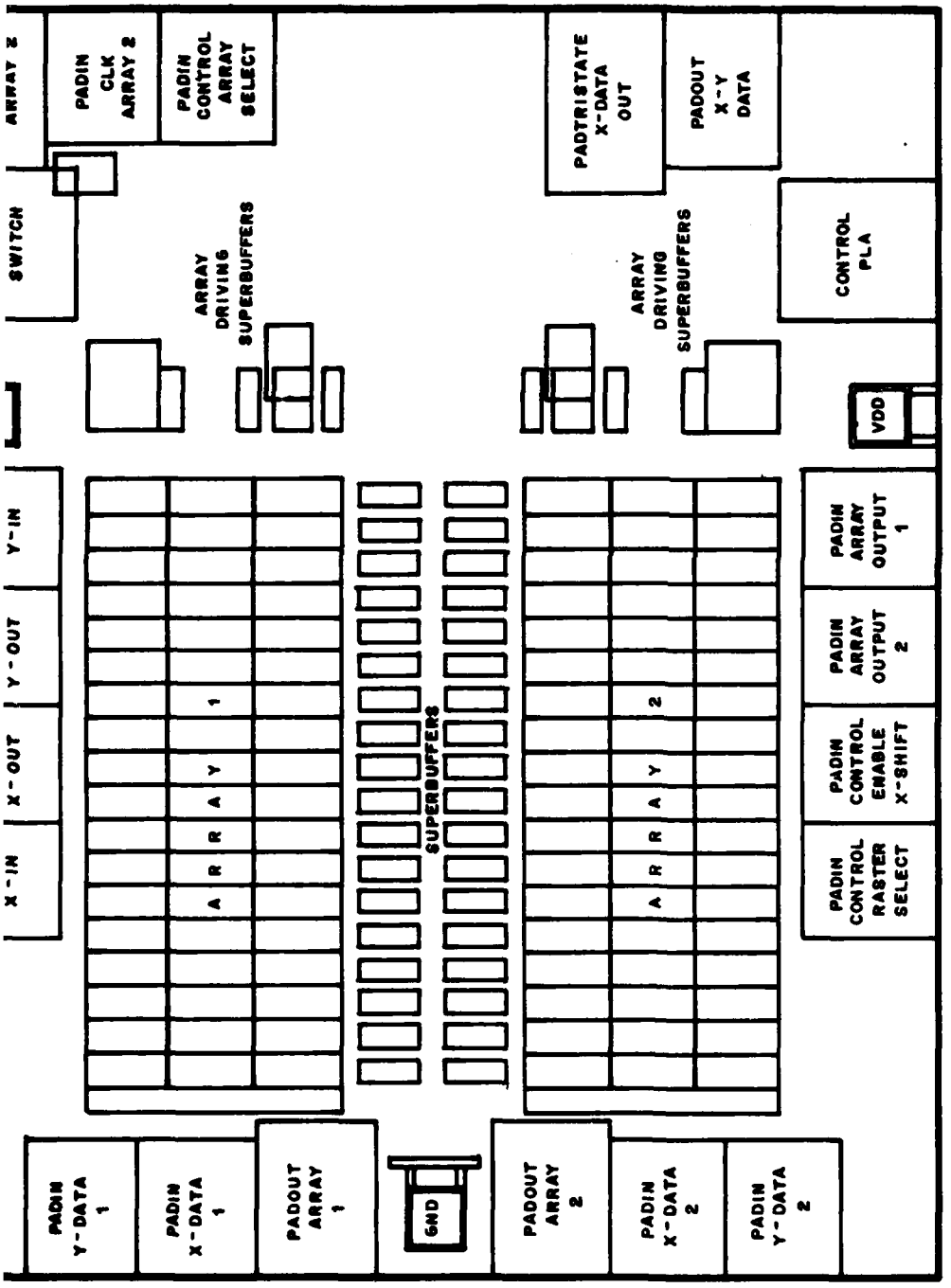


Figure 9. Floor plan of project

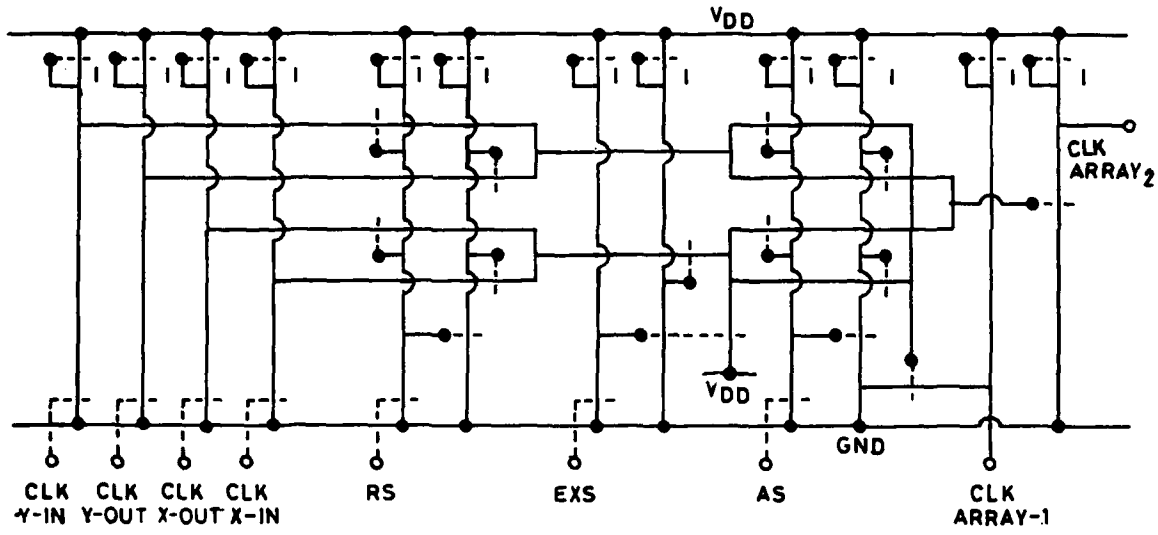
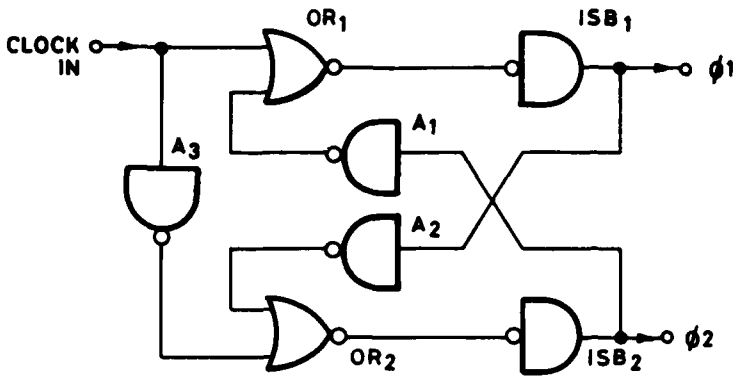
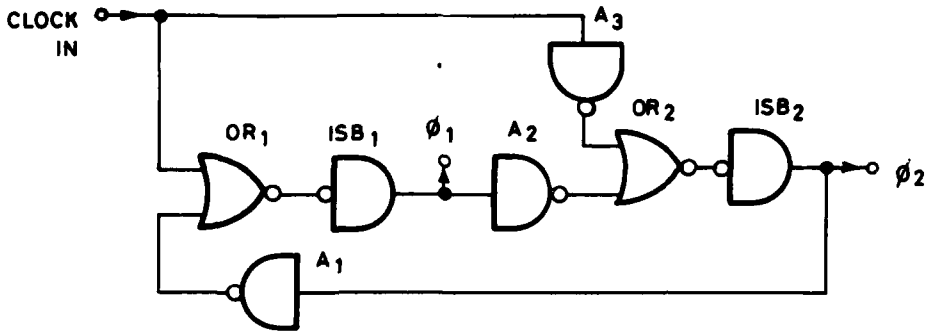


Figure 10. Clock Selector Switch - stick diagram

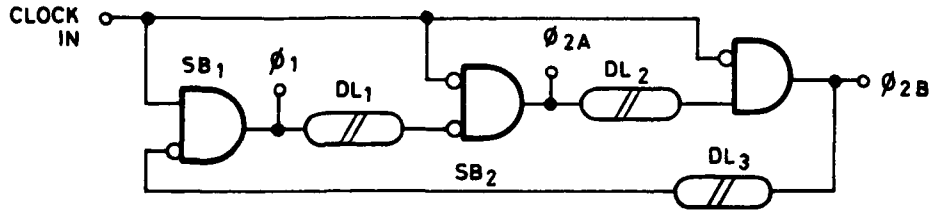


(a) logic diagram

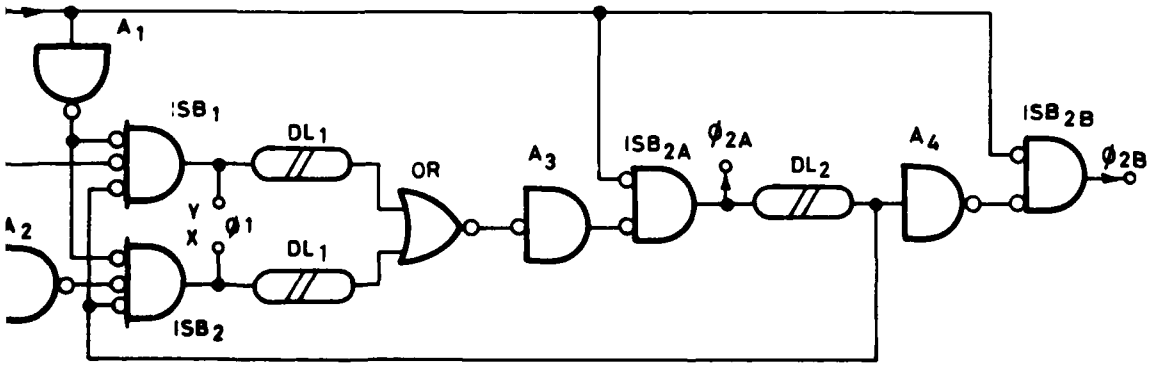


(b) rearranged logic diagram

Figure 11. Standard cell PADCLOCKBAR



(a) basic logic diagram



(b) actual logic diagram

Figure 12. Clock Splitter

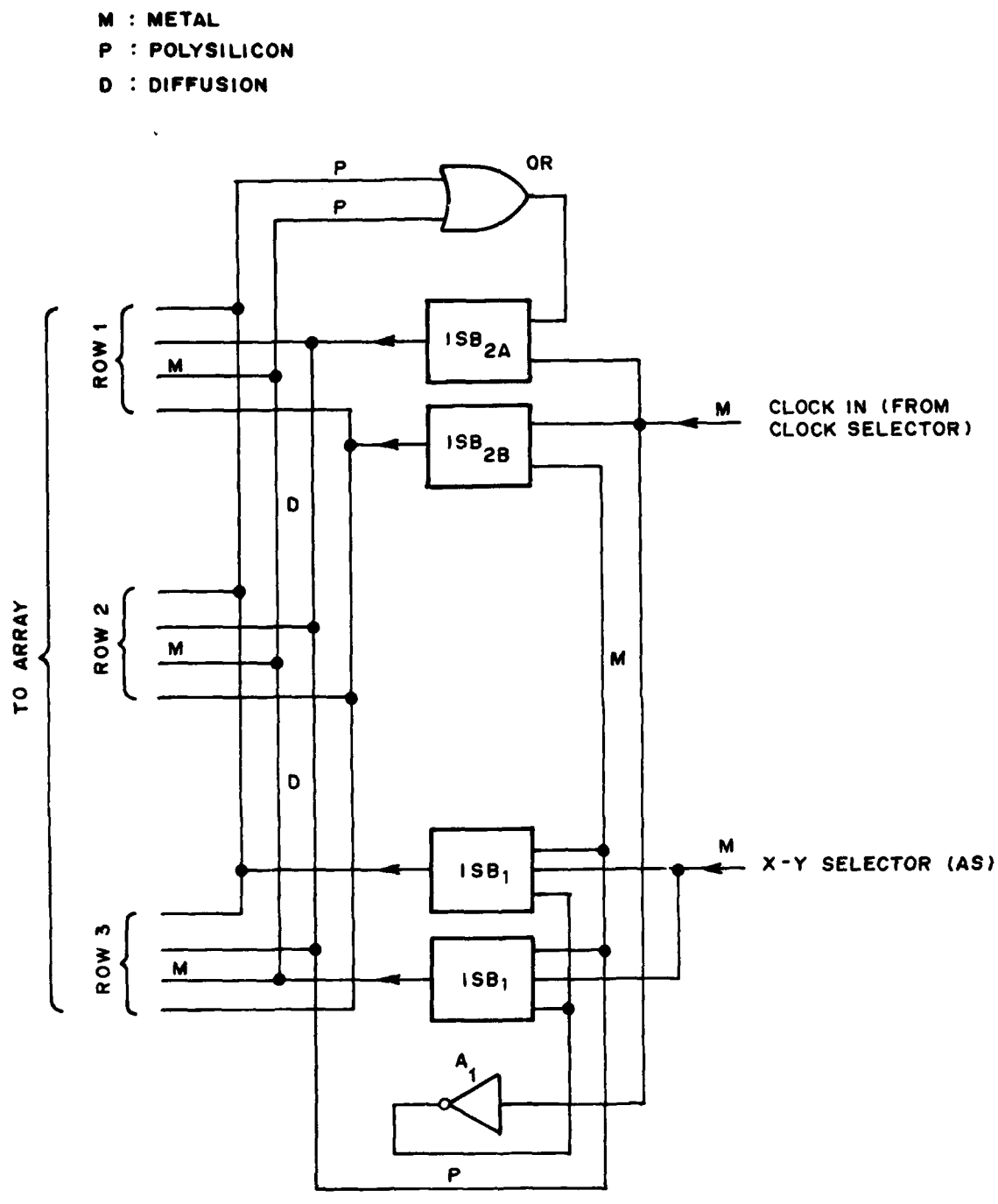


Figure 13. Clock Splitter physical arrangement

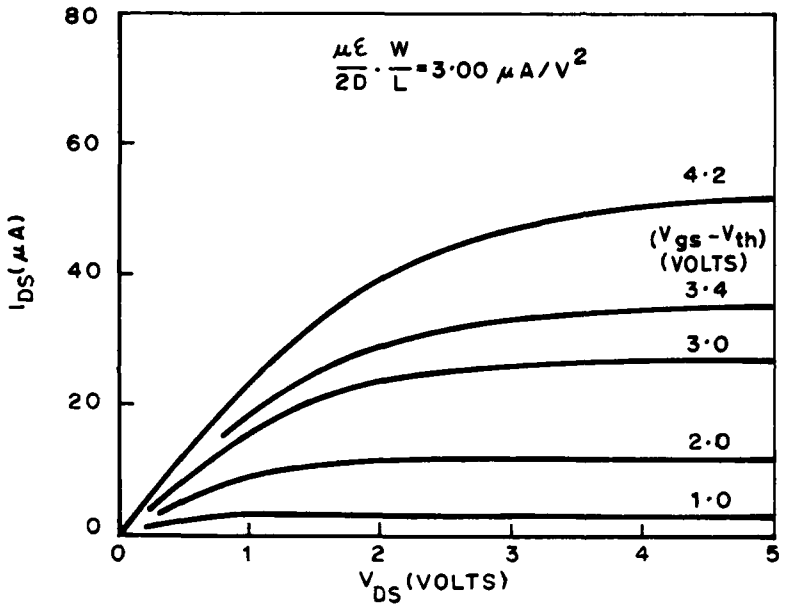


Figure 14. FET transistor model

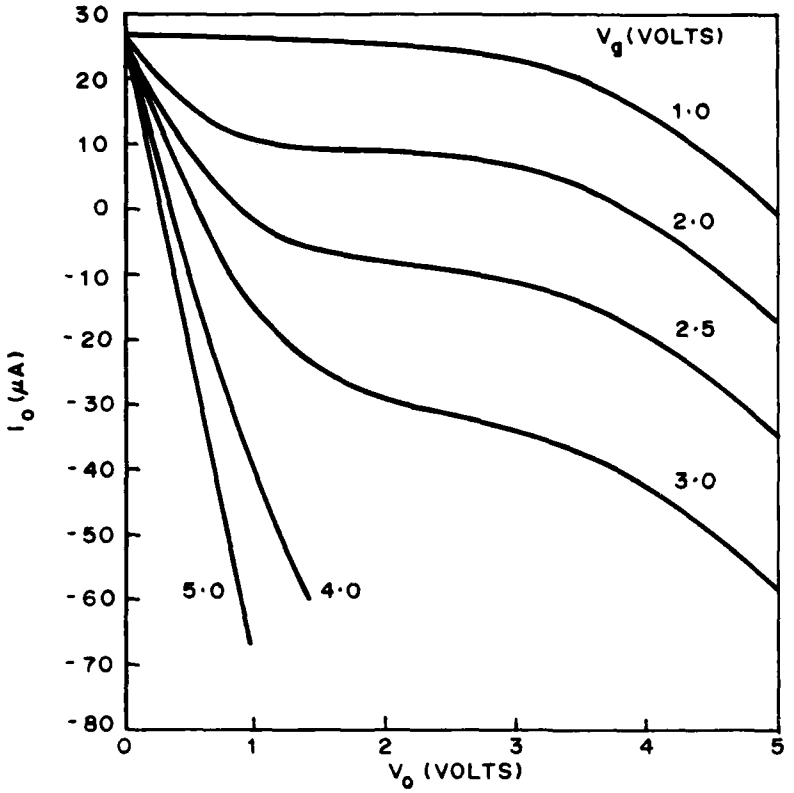


Figure 15. FET gate model (R = 4)

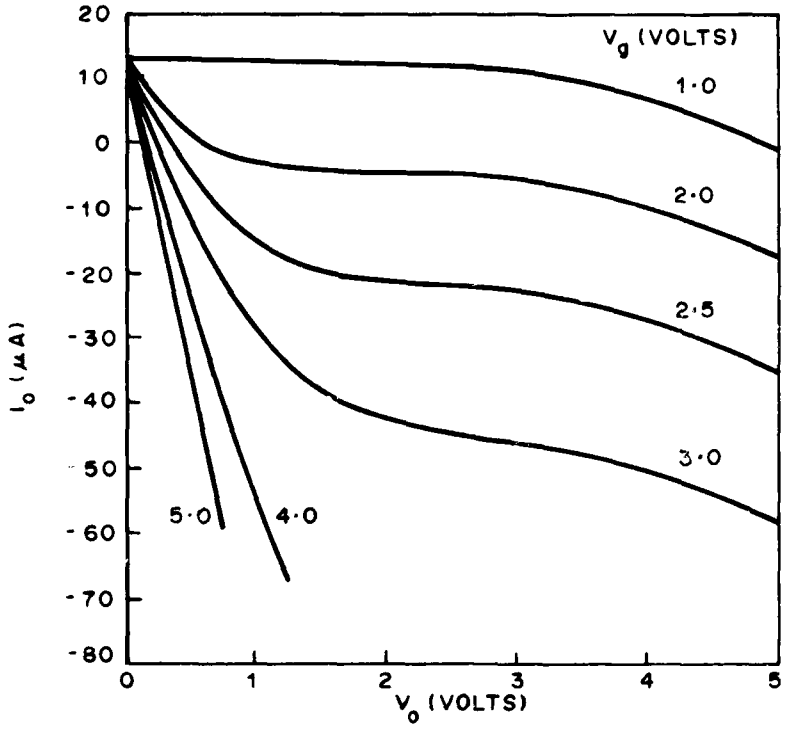
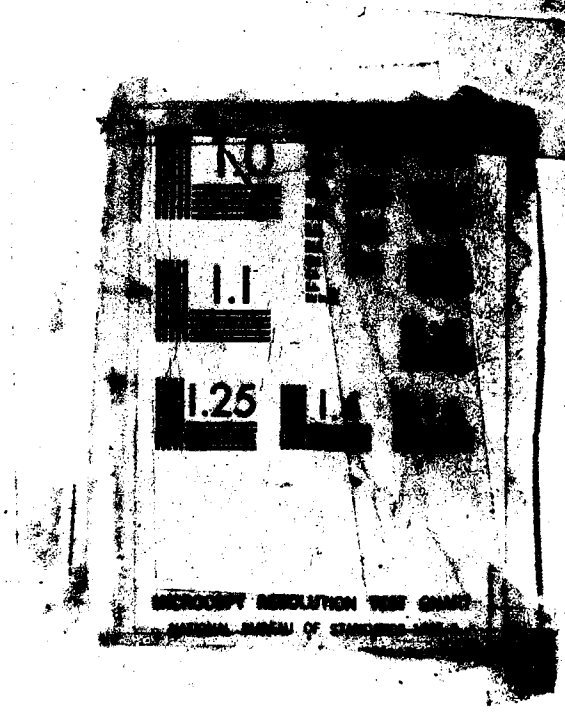
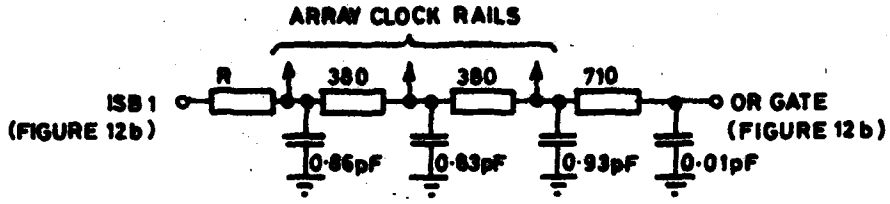


Figure 16. FET gate model (R = 8)

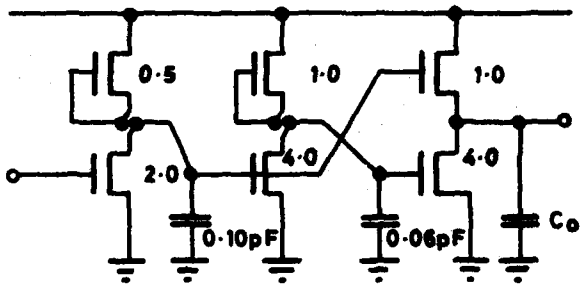
AD-A157 129 VLSI WORKSHOP AND PROJECT: FEBRUARY-NOVEMBER 1982(U) 2/2
WEAPONS SYSTEMS RESEARCH LAB ADELAIDE (AUSTRALIA)
J HAYWARD SEP 84 WSRL-0374-TR

UNCLASSIFIED F/G 9/5 NL

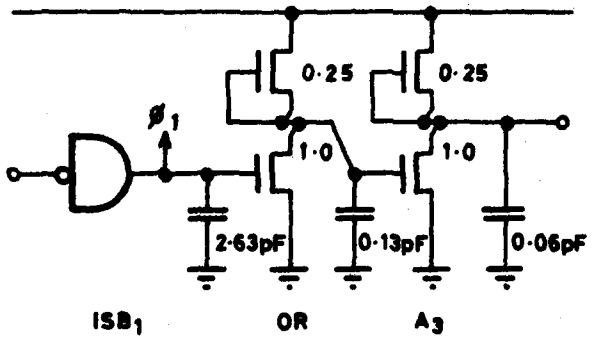

1.1
1.25



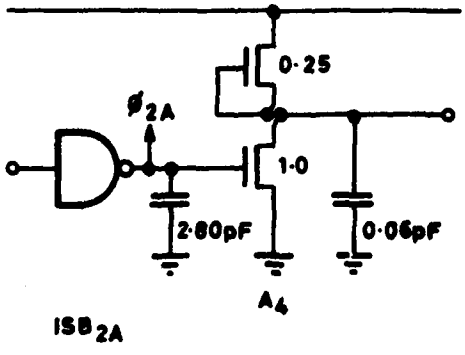
(a) tie-bus equivalent circuit



(b) inverting super buffer



(c) phase 1 generator



(d) phase 2a generator

Figure 17. Clock Splitter circuits

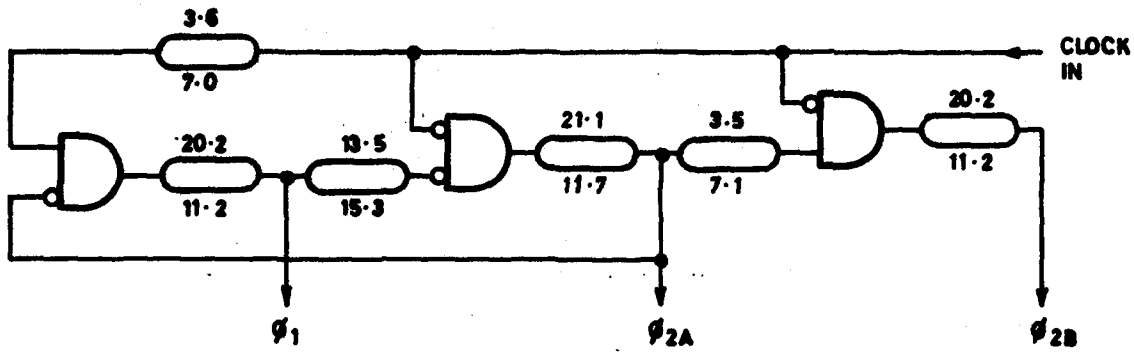
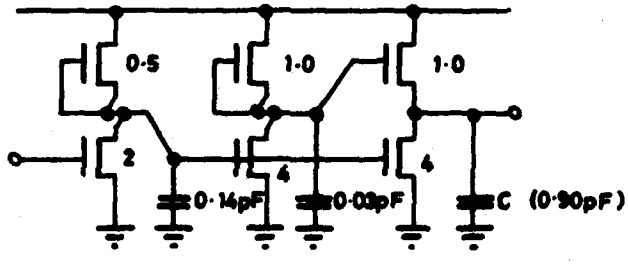
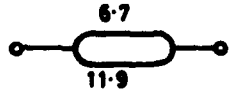


Figure 18. Clock Splitter estimated delays



(a) equivalent circuit



(b) effective time delay

Figure 19. Non-Inverting Super-Buffer

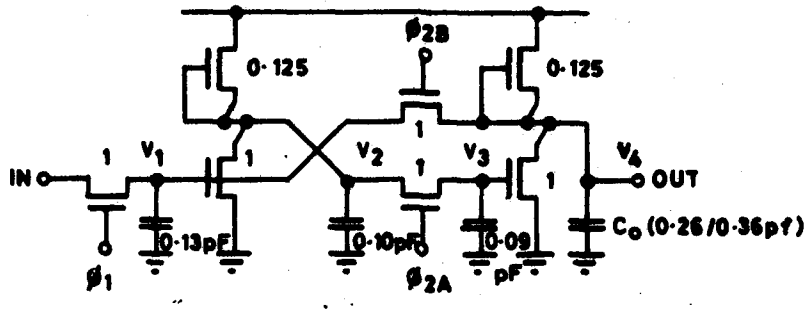


Figure 20. Equivalent circuit of array cell

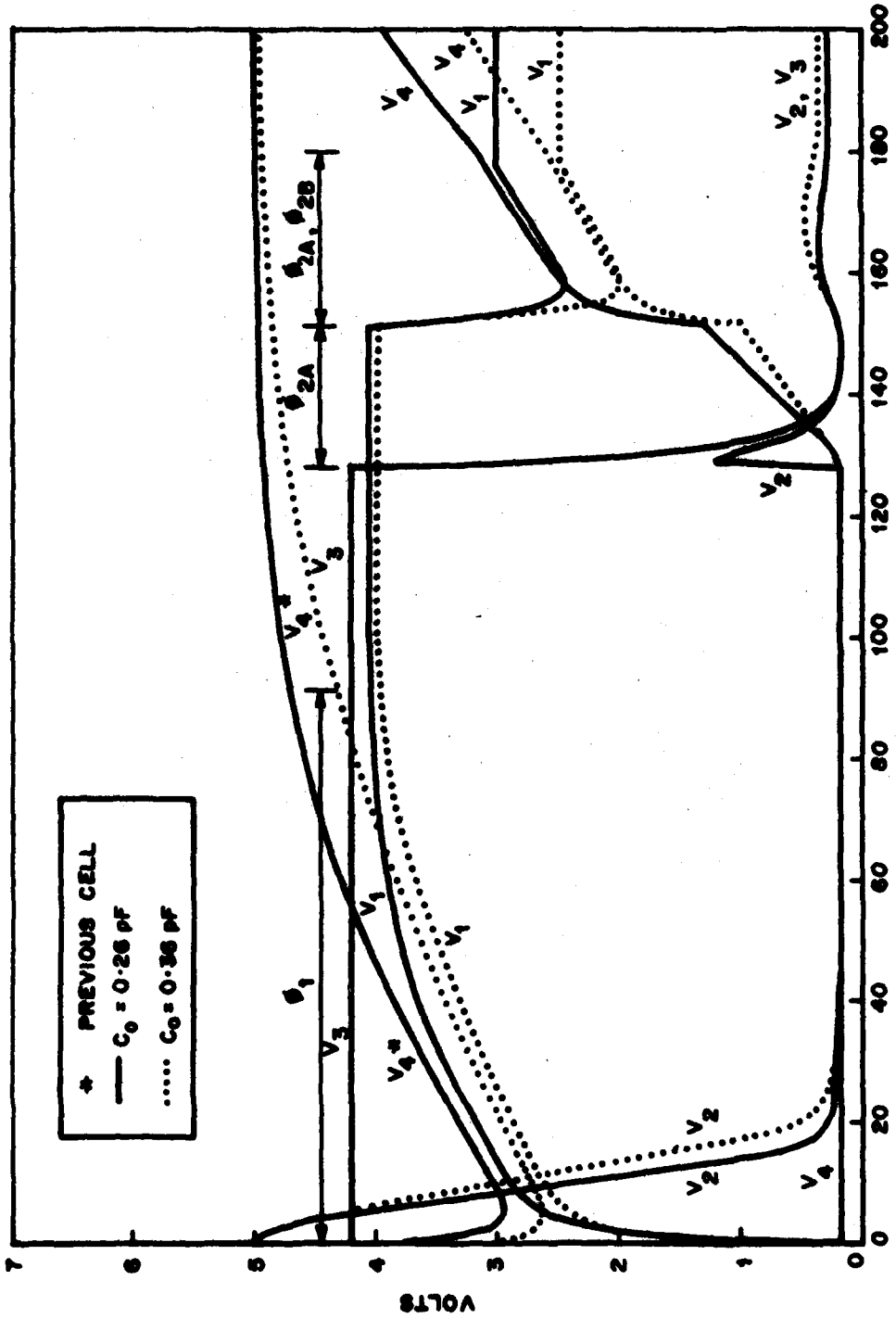


Figure 21. Simulation of shift register transient (0 to 1)

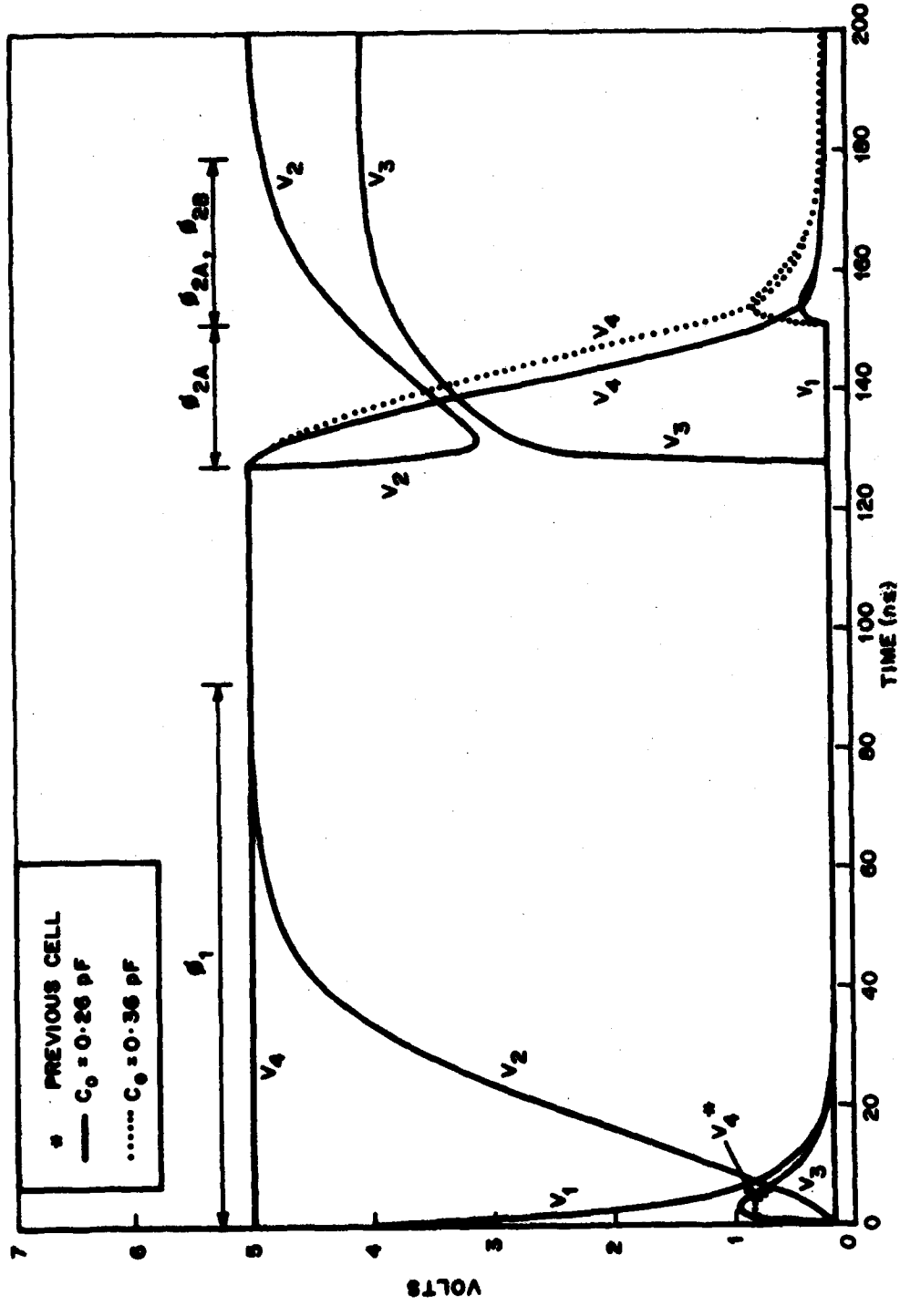
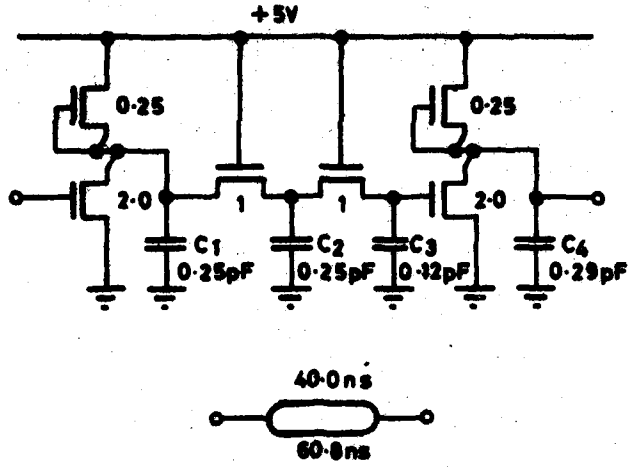
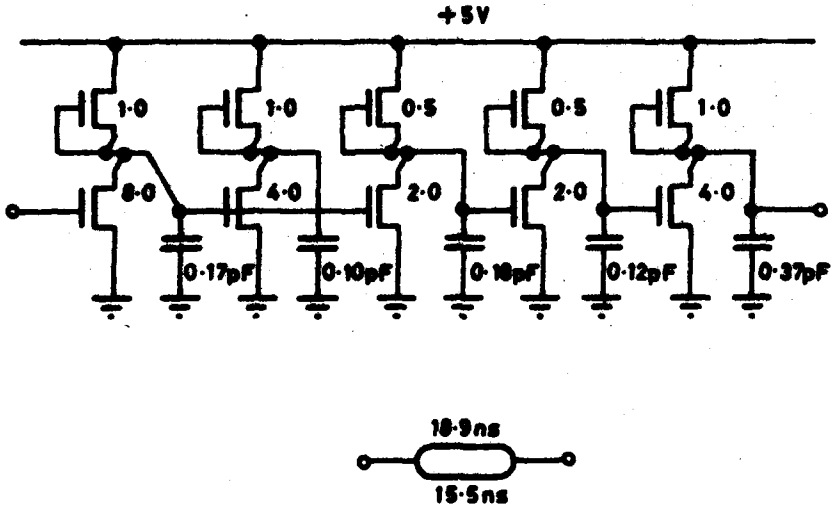


Figure 22. Simulation of shift register transient (1 to 0)



(a) Clock Selector Switch



(b) output PLA

Figure 23. Equivalent circuits

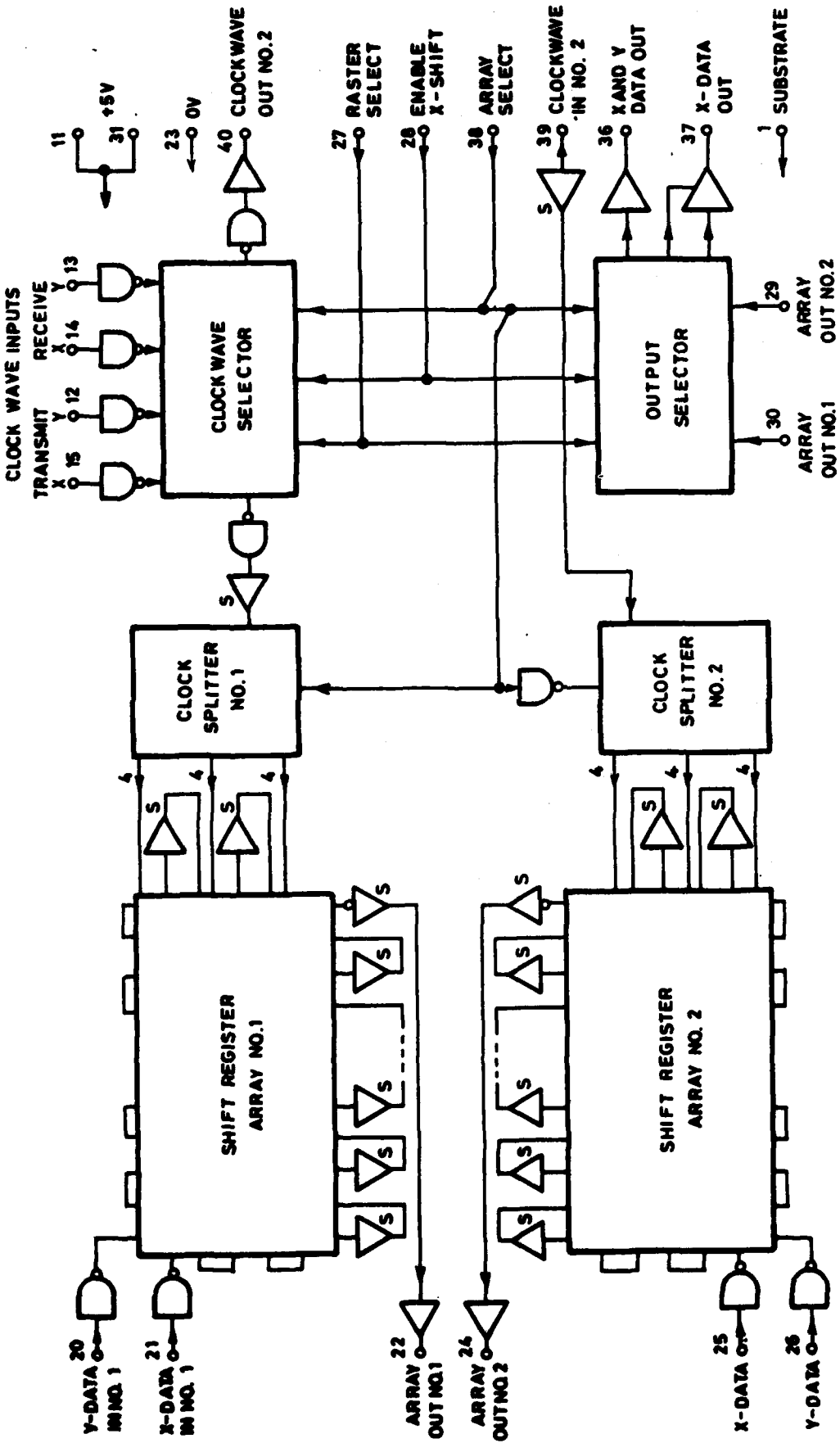


Figure 24. Chip connections and internal logic

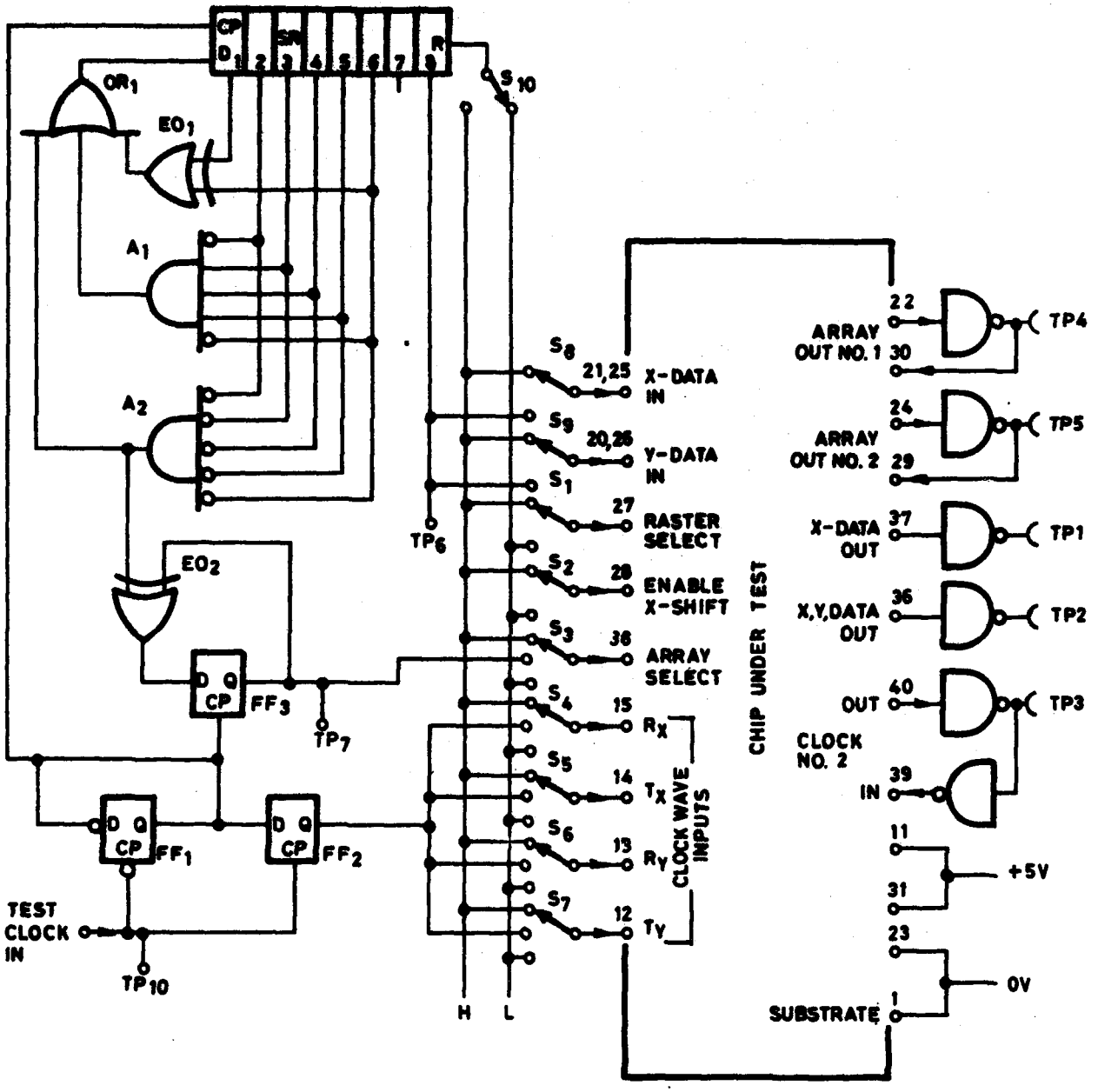


Figure 25. Logic diagram of functional test set up

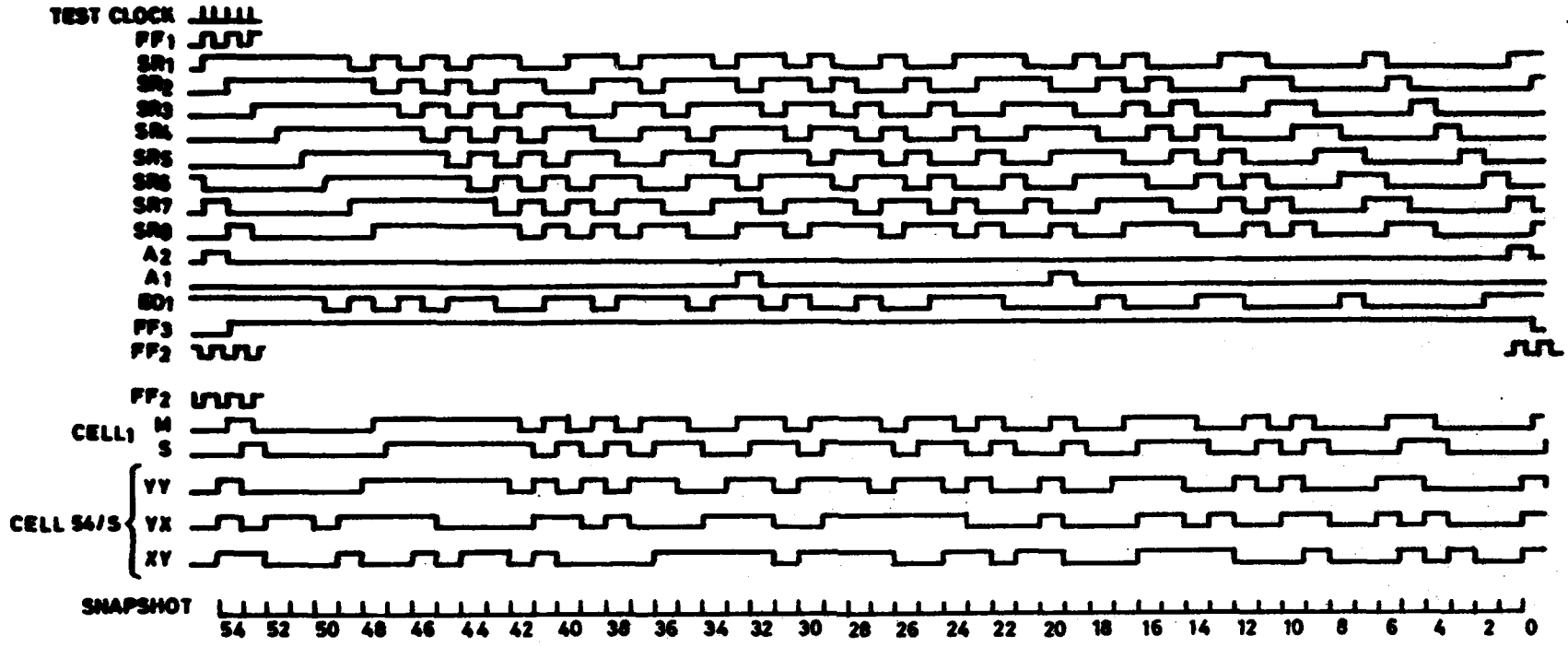


Figure 26. Waveforms for functional test set up

DISTRIBUTION

	Copy No.
DEPARTMENT OF DEFENCE	
Defence Science and Technology Organisation	
Chief Defence Scientist	1
Deputy Chief Defence Scientist	
Controller, External Relations, Projects and Analytical Studies	
Superintendent, Science Programs and Administration	
Defence Science Representative, London	Cnt Sht Only
Counsellor, Defence Science Washington	Cnt Sht Only
Deputy Secretary (Manufacturing)	2
Deputy Secretary (Materiel and Resources)	
Controller, Defence Aerospace Division	
Controller, Munitions Division	
Library, DDS Central Office	3
Director, Industry Development, SA/NT	Cnt Sht Only
Weapons Systems Research Laboratory	
Director, Weapons Systems Research Laboratory	4
Superintendent, Weapon Systems Division	5
Principal Officer, Tracking and Command Systems Group	6
Principal Officer, Systems Integration Group	7
Principal Officer, Systems Modelling Group	8
Principal Officer, Terminal Guidance Group	9
J. Hayward, Tracking and Command Systems Group	10
Electronics Research Laboratory	
Director, Electronics Research Laboratory	11
Superintendent, Radar Division	12
Superintendent, Electronic Warfare Division	13

	Copy No.
Principal Officer, Cybernetic Electronics Group	14
M. Raymond, Communications Electronic Warfare Group	15
A. Doolette, Electronic Warfare Studies Group	16
ranced Engineering Laboratory	
Director, Advanced Engineering Laboratory	17
D. Gapp, Communications Technology Group	18
ronautical Research Laboratories	
Director, Aeronautical Research Laboratories	19
Superintendent, Systems Division	20
terials Research Laboratories	
Director, Materials Research Laboratories	21
vernment Aircraft Factories	
Manager/Library	22
Chief Designer	23
vy Office	
Navy Scientific Adviser	Cnt Sht Only
Director, Naval Weapon Design	24
my Office	
Director General, Army Development (NSO), Russell Offices for ABCA Standardisation Officers	
UK ABCA representative, Canberra	25
US ABCA representative, Canberra	26
Canada ABCA representative, Canberra	27
NZ ABCA representative, Canberra	28
braries and Information Services	
Librarian Technical Reports Centre, Defence Central Campbell Park	29

Copy No.

Document Exchange Centre
Defence Information Services Branch for:

Microfilming	30
United Kingdom, Defence Research Information Centre (DRIC)	31 - 32
United States, Defense Technical Information Center	33 - 44
Canada, Director, Scientific Information Services	45
New Zealand, Ministry of Defence	46
National Library of Australia	47
Director, Joint Intelligence Organisation	48
Library, Defence Research Centre Salisbury	49 - 50
Library, Aeronautical Research Laboratories	51
Library, Materials Research Laboratories	52
Library, Royal Australian Navy Research Laboratories	53
Library, H Block, Victoria Barracks, Melbourne	54
UNITED KINGDOM	
British Library, Lending Division	55
UNITED STATES OF AMERICA	
NASA Scientific and Technical Information Office	56
Engineering Societies Library	57
Spares	58 - 62

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	2 SECURITY CLASSIFICATION
AR Number: AR-004-105	a. Complete Document: Unclassified
Series Number: WSRL-0374-TR	b. Title in Isolation: Unclassified
Other Numbers:	c. Summary in Isolation: Unclassified

3 TITLE

FINAL REPORT ON VLSI WORKSHOP AND PROJECT: FEBRUARY - NOVEMBER 1982

4 PERSONAL AUTHOR(S):

J. Hayward

5 DOCUMENT DATE:

September 1984

6 6.1 TOTAL NUMBER OF PAGES 97

6.2 NUMBER OF REFERENCES: 5

7 7.1 CORPORATE AUTHOR(S):

Weapons Systems Research Laboratory

7.2 DOCUMENT SERIES AND NUMBER

Weapons Systems Research Laboratory 0374-TR

8 REFERENCE NUMBERS

a. Task:

b. Sponsoring Agency:

9 COST CODE:

451167

10 IMPRINT (Publishing organisation)

Defence Research Centre Salisbury

11 COMPUTER PROGRAM(S) (Title(s) and language(s))

12 RELEASE LIMITATIONS (of the document):

Approved for Public Release

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

No Limitation

14 DESCRIPTORS:

a. EJC Thesaurus Terms

Integrated circuits;
Chips (electronics);
Computer aided design;
Design
Manufacture. (Australia).

b. Non-Thesaurus Terms

VLSI

15 COSATI CODES:

09010

16 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

At regular intervals, the CSIRO VLSI Program coordinates the production of multiproject very large scale integrated circuit chips, in n-mos technology, using the Mead-Conway design method. The author attended the first Workshop conducted in support of this activity, and designed a successful digital circuit project which became part of the first such chip. This report describes the Workshop, the project, the circuit design, the performance estimation and measured performance. It also covers the difficulties encountered in transfering the computer software design tools, supplied by the Program in a form suitable for the VAX computer operating system, to the main-frame IBM 3033 system in use at DRCS. Keywords:

Security classification of this page: UNCLASSIFIED

The official documents produced by the Laboratories of the Defence Research Centre Salisbury are issued in one of five categories: Reports, Technical Reports, Technical Memoranda, Manuals and Specifications. The purpose of the latter two categories is self-evident, with the other three categories being used for the following purposes:

Reports	:	documents prepared for managerial purposes.
Technical Reports	:	records of scientific and technical work of a permanent value intended for other scientists and technologists working in the field.
Technical Memoranda	:	intended primarily for disseminating information within the DSTO. They are usually tentative in nature and reflect the personal views of the author.

END

FILMED

9-85

DTIC